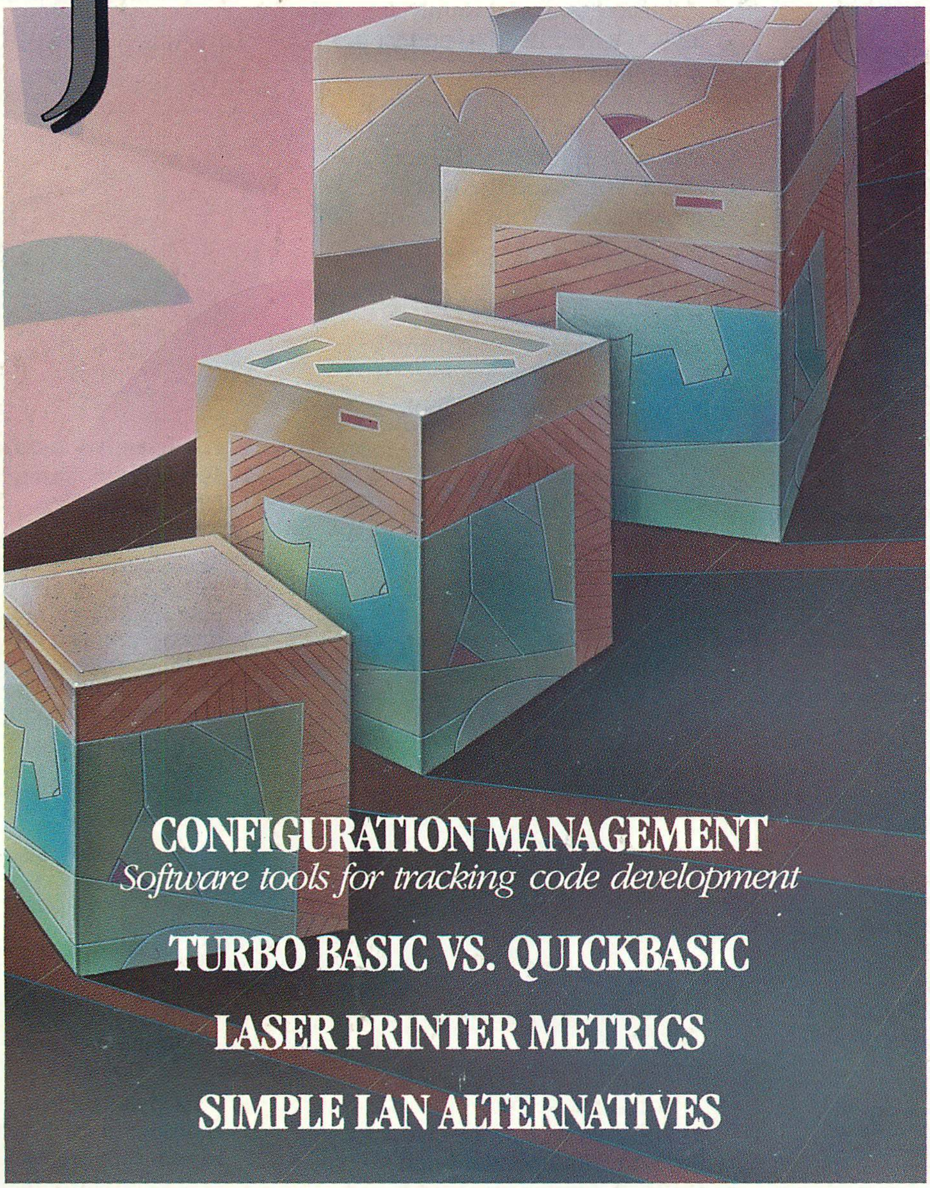
SEPTEMBER 1987 VOL. 5 NO. 9 $3.95

FOR THE SYSTEMS PROFESSIONAL

PC TECH JOURNAL®



CONFIGURATION MANAGEMENT
Software tools for tracking code development

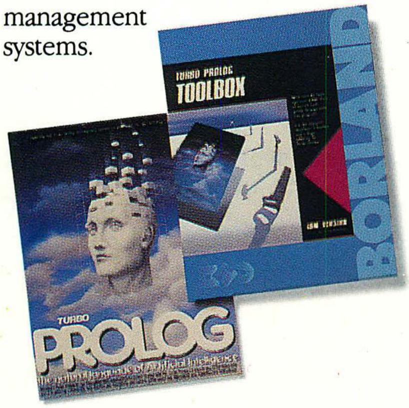
TURBO BASIC VS. QUICKBASIC

LASER PRINTER METRICS

SIMPLE LAN ALTERNATIVES

Turbo Prolog: The Natural Language of Artificial Intelligence

Whether you're a first-time programmer or an experienced one, Turbo Prolog's natural implementation of Artificial Intelligence soon shows you how to build expert systems, natural language interfaces, customized knowledge bases and smart information management systems.



Turbo Prolog and Turbo C work hand-in-hand

Turbo Prolog® interfaces perfectly with Turbo C® because they're both designed to work with each other.

The Turbo Prolog/Turbo C combination means that you can now build powerful commercial applications using two of the most powerful languages available.

Turbo Prolog's development system includes:

- ☑ A complete Prolog compiler that is a variation of the Clocksin and Mellish Edinburgh standard Prolog.
- ☑ A full-screen interactive editor.
- ☑ Support for both graphic and text windows.
- ☑ All the tools that let you build your own expert systems and **AI** applications with unprecedented ease.

> An affordable, fast, and easy-to-use language that will delight the newcomer . . . You experienced Prolog hackers will likewise be delighted, if not astonished, by the features and performance of the Turbo Prolog development environment.
>
> Turbo Prolog offers generally the fastest and most approachable implementation of that language.
>
> *Darryl Rubin, AI Expert*

How Turbo Prolog's new Toolbox adds 80 powerful tools and 8000 lines of source code

In keeping with Borland tradition, we've quickly added the new Turbo Prolog Toolbox™ to Turbo Prolog.

With 80 tools and 8000 lines of source code that can easily be incorporated into your own programs—and 40 sample programs that show you how to put these AI tools to work—the Turbo Prolog Toolbox is a highly intelligent, high-performance addition.

Only $99.95!

Turbo Prolog Toolbox features include:

- ☑ Business graphics generation: boxes, circles, ellipses, bar charts, pie charts, scaled graphics
- ☑ Complete communications package: supports XMODEM protocol
- ☑ File transfers from Reflex,® dBASE III,® 1-2-3,® Symphony®
- ☑ A unique parser generator: construct your own compiler or query language
- ☑ Sophisticated user-interface design tools
- ☑ Contains 40 example programs
- ☑ Easy-to-use screen editor: design your screen layout and I/O
- ☑ Calculated fields definition
- ☑ Over 8,000 lines of source code you can incorporate into your own programs

Turb
The most powe
compil

Our new Turbo C generates fast, tight, production-quality code at compilation speeds of more than 13,000 lines a minute!

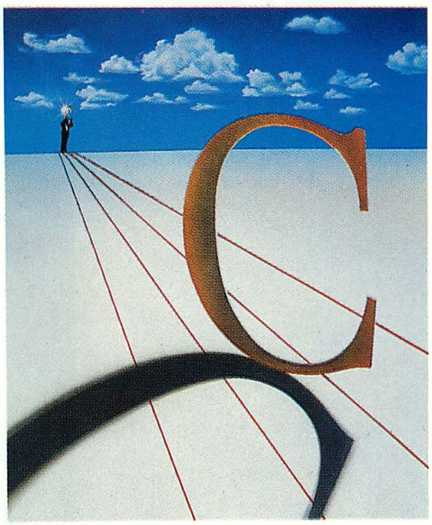
It's the full-featured optimizing compiler everyone has been waiting for.

Switching to Turbo C, or starting with Turbo C, you win both ways

If you're already programming in C, switching to Turbo C will make you feel like you're riding a rocket instead of pedaling a bike.

If you're never programmed in C, starting with Turbo C gives you an instant edge. It's easy to learn, easy to use, and the most efficient C compiler at any price.

Only $99.95!



> Turbo C does look like What We've All Been Waiting For: a full-featured compiler that produces excellent code in an unbelievable hurry . . . moves into a class all its own among full-featured C compilers . . . Turbo C is indeed for the serious developer . . . One heck of a buy—at any price.
>
> *Michael Abrash, Programmer's Journal*

o C: NEW!
rful optimizing
r ever

Sieve benchmark

	Turbo C	Microsoft® C
Compile time	*2.4*	13.51
Compile and link time	*4.1*	18.13
Execution time	*3.95*	5.93
Object code size	*239*	249
Execution size	*5748*	7136
Price	*$99.95*	$450.00

Benchmark run on an IBM PS/2 Model 60 using Turbo C version 1.0 and the Turbo Linker version 1.0; Microsoft C version 4.0 and the MS overlay linker version 3.51.

Technical Specifications

- ☑ Compiler: One-pass optimizing compiler generating linkable object modules. Included is Borland's high-performance Turbo Linker.™ The object module is compatible with the PC-DOS linker. Supports tiny, small, compact, medium, large, and huge memory model libraries. Can mix models with near and far pointers. Includes floating point emulator (utilizes 8087/80287 if installed).
- ☑ Interactive Editor: The system includes a powerful, interactive full-screen text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code.
- ☑ Development Environment: A powerful "Make" is included so that managing Turbo C program development is highly efficient. Also includes pull-down menus and windows.
- ☑ Links with relocatable object modules created using Borland's Turbo Prolog into a single program.
- ☑ Inline assembly code.
- ☑ Loop optimizations.
- ☑ Register variables.
- ☑ ANSI C compatible.
- ☑ Start-up routine source code included.
- ☑ Both command line and integrated environment versions included.
- ☑ License to the source code for Runtime Library available.

Join more than 100,000 Turbo C enthusiasts. Get your copy of Turbo C today!

Minimum system requirements: All products run on IBM PC, XT, AT, PS/2, portable and true compatibles. PC-DOS (MS-DOS) 2.0 or later. 384K RAM minimum. Basic Telecom and Editor Toolboxes require 640K.

Borland International
4585 Scotts Valley Drive, Scotts Valley, CA 95066
Telephone: (408) 438-8400 Telex: 172373

Why more than 600,000 programmers worldwide are using Turbo Pascal today

The irresistible force behind Turbo Pascal's worldwide success is Borland's advanced technology. We created a compiler so fast, that Turbo Pascal® is now the worldwide standard. And there are more tools for Turbo Pascal than for any other development environment in the world.

You'll get everything you need from Turbo Pascal and its 5 Toolboxes

Turbo Pascal and Family are all you'll ever need to perfect programming in Pascal.

If you've never programmed in Pascal, you'll probably want to start with Turbo Pascal Tutor® 2.0, and as your expertise quickly grows, add Toolboxes like our

- Database Toolbox®
- Editor Toolbox®
- Graphix Toolbox®
- GameWorks®
  and our newest,
- Numerical Methods Toolbox™

And because Turbo Pascal is the established worldwide standard, 3rd party, independent non-Borland developers also offer an incredible array of programs for Turbo Pascal. ***Only $99.95!***

> "Borland International's Turbo Pascal took the programming world by storm. A great compiler combined with a good editor at an astounding price, the package quickly came to be called, simply, Turbo—and has sold more than 500,000 copies.
>
> *Stephen Randy Davis, PC Magazine*
>
> Language deal of the century. *PC Magazine*"

For Scientists and Engineers: Turbo Pascal Numerical Methods Toolbox

The Numerical Methods Toolbox is a complete collection of Turbo Pascal routines and programs. Add it to your development system and you have the most comprehensive and powerful numerical analysis capabilities—at your fingertips!

The Numerical Methods Toolbox is a state-of-the-art mathematical toolbox with these ten powerful features:

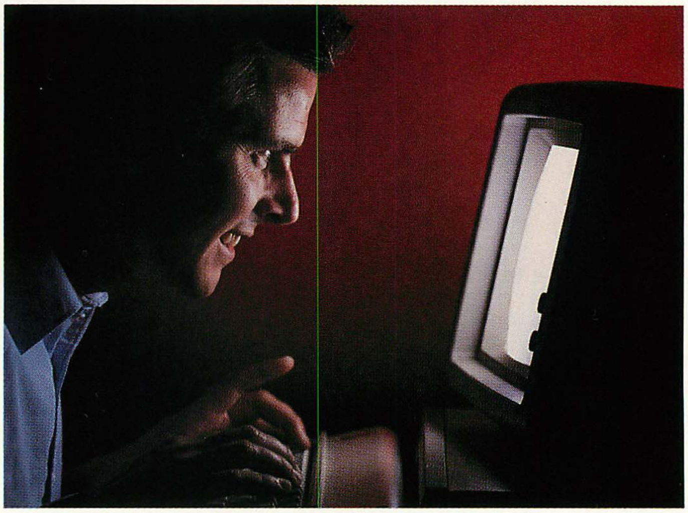
- ☑ Zeros of a function
- ☑ Interpolation
- ☑ Differentiation
- ☑ Integration
- ☑ Matrix Inversion
- ☑ Matrix Eigenvalues
- ☑ Differential Equations
- ☑ Least Squares
- ☑ Fourier Transforms
- ☑ Graphics

Each module comes with procedures that can be easily adapted to your own program. The Toolbox also comes complete with source code. So you have total control of your application.

Only $99.95!

BI-1131A

Turbo C, Turbo Basic, Turbo Pascal and Turbo Prolog: technical excellence



"Borland International's Turbo Pascal, Turbo Basic and Turbo Prolog automatically identify themselves, by virtue of their 'Turbo' forenames, as superior language products with a common programming environment. The appellation also means to many PC users a 'must have' language. To us Turbo C looks like a coup for Borland.

Garry Ray, PC Week"

BI-1131A

Turbo Basic introduces its powerful new Telecom, Editor and Database Toolboxes

NEW!

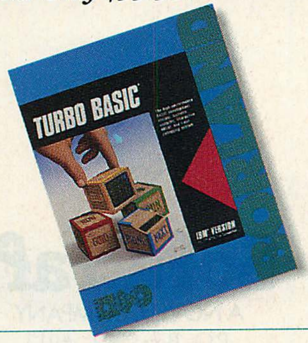
Turbo Basic® is the breakthrough you've been waiting for. The same power we brought to Pascal with Turbo Pascal has now been applied to BASIC with Turbo Basic.

Compatible with BASICA, Turbo Basic is the high-performance, high-speed BASIC you'd expect from Borland.

Basically, Turbo Basic is all you need

It's a complete development environment which includes an incredibly fast compiler, an interactive editor and a trace debugging system. It outperforms all its rivals, and because it's compatible with BASICA, you probably already know how to use it.

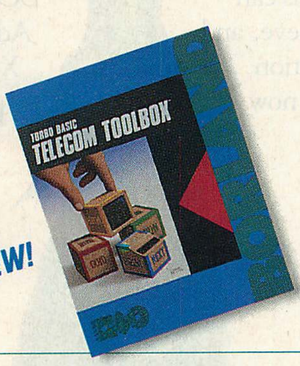
Includes a free MicroCalc™ spreadsheet complete with source code. ***Only $99.95!***



A technical look at Turbo Basic

- ☑ Full recursion supported
- ☑ Standard IEEE floating-point format
- ☑ Floating-point support, with full 8087 (math co-processor) integration. Software emulation if no 8087 present
- ☑ Program size limited only by available memory (no 64K limitation)
- ☑ VGA, CGA, and EGA support
- ☑ Access to local, static, and global variables
- ☑ Full integration of the compiler, editor, and executable program, with separate windows for editing, messages, tracing, and execution
- ☑ Compile, run-time, and I/O errors place you in the source code where error occurred
- ☑ New long integer (32-bit) data type
- ☑ Full 80-bit precision
- ☑ Pull-down menus
- ☑ Full window management

> "Borland has created the most powerful version of BASIC ever."
>
> *Ethan Winer, PC Magazine*



NEW!

Telecom Toolbox is a complete communications package which takes advantage of the built-in communications capabilities of BASIC—use as is or modify.

- Pull-down menus and windows
- XMODEM support
- VT 100 terminal emulation
- Captures text to disk or printer
- PhoneBook file
- 300, 1200, 2400 baud support
- Supports script files
- Fast screen I/O
- Supports most of XTalk's command set
- Manual dial and redial options

Use Telecom Toolbox to embed communications capabilities into your own programs and/or build your own communications package. Source code included for all Toolbox code and sample programs. ***Only $99.95!***

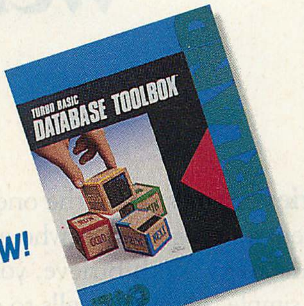
For the dealer nearest you or to order by phone call

(800) 255-8008

in CA (800) 742-1133 in Canada (800) 237-1136

BORLAND INTERNATIONAL

SUMMER BREAK SPECIAL!
Buy Turbo Basic and get a **FREE** product.
See your dealer for details!

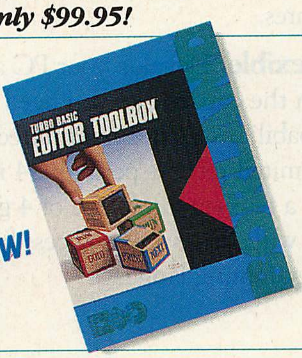


NEW!

Database Toolbox means that you don't have to reinvent the wheel each time you write new Turbo Basic database programs.

- ☑ "Trainer" shows you how B+ trees work. (Simply key in sample records and you'll see your index being built.)
- ☑ Turbo Access instantly locates, inserts or deletes records in a database—using B+ trees.
- ☑ Turbo Sort sorts data on single items or on multiple keys and features virtual memory management for sorting large data files.

Source code included.
Only $99.95!



NEW!

Editor Toolbox is all you need to build your own text editor or word processor. Includes source code for two sample editors.

First Editor is a complete editor ready to include in your programs, complete with windows, block commands and memory-mapped screen routines.

MicroStar™ is a full-blown text editor with a complete pull-down menu user interface, and gives you

- Wordwrap
- Undo last change
- Auto-Indent
- Find and Find/Replace with options
- Set left/right margins
- Block mark, move and copy
- Tab, insert, overstrike modes, line center etc.

Includes source code.
Only $99.95!

BI-1131A

CIRCLE NO. 254 ON READER SERVICE CARD

If you thought all file managers were alike, think again. Think Btrieve.

Btrieve stands out as the one file management system for programmers who are serious about productivity. With Btrieve, your programs can use simple subroutine calls to store, retrieve, and update records. It's the file handling solution rich in functionality to meet your needs now and in the future, as your requirements grow and change.

Bfast. With Btrieve you develop fast applications, with ease. Written in assembly language for IBM PC's, Btrieve uses b-tree algorithms with automatic balancing for fast, efficient file indexing.

Bsafe. Btrieve is the only file manager with automatic file recovery. Two levels of database integrity provide complete fault tolerance in the event of accidents or power failures.

Bflexible. Develop your PC applications with the capabilities you need most. Capabilities such as: unlimited open files, unlimited records per file, 24 indexes per file and a maximum file size of 4 gigabytes. All this with Btrieve plus access from any programming language.

Bconnected. When your applications require networking, think Btrieve. A single version runs on all DOS 3 LAN's, including IBM PC Network and Novell Advanced NetWare. Btrieve is also available for Xenix and other multi-tasking operating systems.

Broyalty Free. No royalty payments on your applications. Ever.

Bsure. Btrieve comes with comprehensive documentation and unrivaled technical support for trouble-free software development. And you can add Xtrieve to supply your end users with even more, a menu-driven report writer and query system.

Think Btrieve. Make your programming performance stand out. Contact SoftCraft.

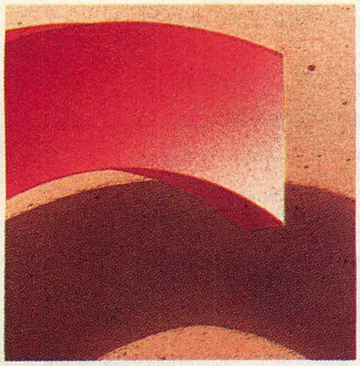
SoftCraft
A NOVELL COMPANY
P.O. Box 9802, #917
Austin, Texas 78766
(512) 346-8380 Telex 358 200

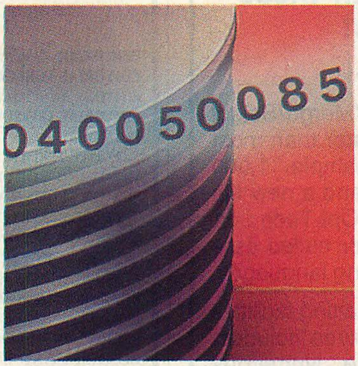
Btrieve, $245; multi-user Btrieve, $595; Xtrieve, $245; multi-user Xtrieve, $595 (for report generation, add $145 for single-user and $345 for multi-user). Requires PC-DOS or MS-DOS 2.X, 3.X or Xenix. Btrieve and Xtrieve are registered trademarks of SoftCraft Inc.

CIRCLE NO. 201 ON READER SERVICE CARD

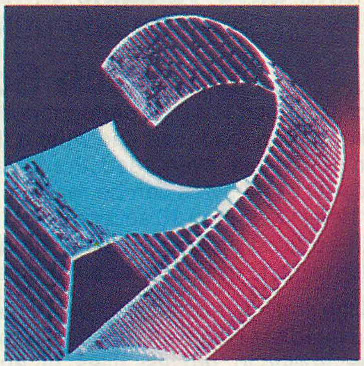
PC TECH JOURNAL

SEPTEMBER 1987 VOLUME 5, NUMBER 9


Succeeding C **166**



Refining Mainframe Access **118**


Laser Metrics **74**

Cover illustration · Andy Levine

PC TECH JOURNAL®

VOL. 5, NO. 9

PUBLISHER: *Newton Barrett*
EDITORIAL DIRECTOR: *Will Fastie*
EDITOR: *Julie Anderson*
EDITORIAL CONSULTANT: *Marjory Spraycar*

EDITORIAL
EXECUTIVE EDITORS: *Susan Holly, David Methvin*
MANAGING EDITOR: *Gail Shaffer*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Caroline Halliday, Ted Mirecki*
ASSOCIATE TECHNICAL EDITOR: *Maxine Fontana*
SENIOR COPY EDITOR: *Bruce Ansley*
NEW PRODUCTS EDITOR: *Carole Eyring*
BUSINESS MANAGER: *Trish Ledbetter*
EDITORIAL SECRETARY: *Valerie Rose*
RECEPTIONIST: *Sharlene Sillaman*
CONTRIBUTING EDITORS: *Steven Armbrust, Dave Browning, Richard M. Foard, Ted Forgeron, Thomas V. Hoffmann, Henry F. Ledgard, Max Stul Oppenheimer, Richard L. Schwartz, Robert E. Shostak*

ART & PRODUCTION
CREATIVE DIRECTOR: *Ina Saltz*
ART DIRECTOR: *Sharon Reuter*
ASSOCIATE ART DIRECTOR: *Courtney Barone*
ASSISTANT ART DIRECTOR: *Crystal Hopkins*
PRODUCTION MANAGER: *Alison Regan Mrohs*
CONTRIBUTING ARTISTS: *Maciek Albrecht, David Povilaitis*

ADVERTISING SALES
NATIONAL SALES MANAGER: *Rita Burke*
ADVERTISING MANAGER/WEST COAST: *Phyllis Egan*
MARKETING DIRECTOR: *Marti Cunha*
SALES SUPPORT ADMINISTRATOR: *Lucy Frazer*
ADVERTISING COORDINATOR: *JeanMarie Donlin*
MARKETING COORDINATOR: *Kim Schroeder*
DISTRICT MANAGERS: *Bill Barney—Southeast; Gayl Sorota—New England; John Blake—Mid-Atlantic; Mike Leonard—Midwest; Steve Moorman, Deborah Gisonni, Nan Hanna—West Coast*
ACCOUNT REPRESENTATIVES: *Mary Martin—Southeast and New England; Nannette Vilushis—Midwest and Mid-Atlantic; Carey Clarke, Patti Ahlswede—West Coast; Classified Advertising Director—Kathryn Cumberlander*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*
RETAIL SALES MANAGER: *Carol Benedetto*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT, Marketing: *Paul H. Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Research: *Marian O. White*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
CHAIRMAN: *Philip B. Korsant;* PRESIDENT: *Kenneth H. Koppel;* SENIOR VICE PRESIDENT: *Philip Sine;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn I. Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. FAX (group 3): 301/740-8809. MCIMail: PCTECH. PCTECHline: 301/740-8383. Telex: 6502565932 MCI.

ADVERTISING OFFICES
(East Coast/Midwest) Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. (Mid-Atlantic) 118 Palisade Avenue, Suite 5C, Cliffside Park, NJ 07010. 201/945-8833. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing. One Park Avenue, 4th floor, New York, NY 10016.

PC Tech Journal (ISSN 0738-0194) is published by Ziff-Davis Publishing Co., a division of Ziff Communications Co., One Park Ave., New York, NY 10016. Published monthly except semi-monthly in December. Subscription rate is $34.97 for one year (13 issues). Additional postage for Canada and Foreign is $.60/copy or $8.00/year. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to PC Tech Journal, P.O. Box 2968, Boulder, CO 80321.

1985 AWARD FOR BEST COMPUTER MAGAZINE Computer Press Association

How to keep up with

Declarative Programming

Procedural programming is for experts, but there are not enough of them to go around.

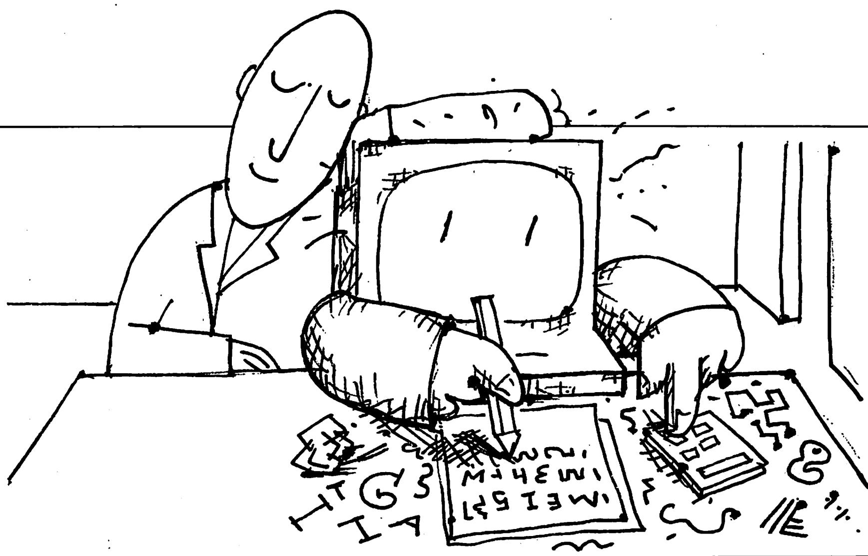


ILLUSTRATION • MACIEK ALBRECHT

I have been spending some time lately with application generators. I mean, I have actually attempted to use some of these products to build small applications—programs I might have written in Pascal or BASIC two years ago.

Although application generators have been around for a long time, purporting to be such productivity gems, most have not lived up to their claims. This time around, however, a couple of these products are truly impressive and may point toward a much brighter future for developers, especially consultants and nonprogrammers. Others are somewhat less impressive, but still represent important levels of utility and productivity for developers.

Sophisticated program generators are important for two reasons. First, professional programmers can be vastly more productive if the computer can do a significant amount of the work, freeing them from the details and allowing them to concentrate on the design and validation of an application.

Second, nonprogrammers trying to mount an application will find it possible to do so, whereas conventional programming might be burdensome.

In both cases, the company who is paying the bills reaps significant rewards: lower project costs and the opportunity to nibble away at the growing applications backlog.

For a while, during the micro revolution, many thought the backlog of applications waiting to be developed diminished. While it did not vanish completely, a lot of pressure came off centralized departments as clever end users figured out ways to solve isolated problems with a tiny computer and a simple program. Data management software then came along for desktops and allowed more sophisticated users to solve more complicated problems. Today, however, the user departments are facing problems far too complex to be solved by developers without a strong technical background. Result: the applications are starting to back up again because there just are not enough technically qualified people to go around. Anything that can break the logjam will be extremely valuable.

TABLES, TABLES, TABLES

One of the two application generators I recently tried and liked is Magic PC, "The Un-Language," from Aker Corporation in Irvine, California. This product extends the well-known data dictionary concept to menus and procedures. The task of creating an application is reduced to filling out tables—for the data structure of each file, the hierarchy of menus, and the actions taken as a result of making a menu choice. (*PC Tech Journal* will review Magic PC in the October issue.)

Although the data dictionary can sometimes seem procedural in nature, it is especially disconcerting to use if you happen to be a programmer. Programmers must overcome the tendency to write a loop or an IF statement and instead must learn just to add a row to the table that describes an action. However, the declarative style of Magic PC is seductive, as is the ease with which an application can be constructed in piecemeal fashion, in the order the developer desires. In fact, parts of the application can be left undone (stubs) while the developer gets the critical parts up and running.

Best of all, Magic PC provides a facility to *verify* the application. At any time, the developer can ask Magic PC to check the integrity of the application system. Inconsistencies are reported to the developer, who can then make the necessary changes. This is very helpful: the program is fast and thorough at running through all the dictionaries and verifying that they make sense.

Magic PC does have its own set of idiosyncrasies. The developer is limited to Magic PC's style of menus and screens and cannot attach a Magic PC application to other code. Clearly, however, it can save the developer countless hours for a reasonably large category of application types.

MENUS, MENUS, MENUS

The big draw for the new version 2.0 of Paradox is supposed to be its network capabilities. They are, in fact, quite good. However, a tool supplied with Ansa's Paradox called the Personal Programmer (PPROG), along with another standard feature called the Data Entry Toolkit, promises to draw even more attention.

PPROG is a tool (a big tool) that guides the developer through an appli-

cation from beginning to end. As with Magic PC, the developer mostly fills out tables, although sometimes PPROG invokes the Paradox form or report generator with which the developer must be familiar. PPROG is a guardian angel; it watches everything you do, checks validity on the fly, and complains when it finds an inconsistency.

For example, I suggested an erroneous table relationship to PPROG during the construction of a sample application. PPROG quickly reported that there did "not seem to be a one-to-one relationship" between the tables, a requirement for the particular situation. That's a tricky bit of information for the program to know, and it is an example of PPROG's sophistication.

Once the application has been defined, it can be tested within the PPROG environment. If the design is not to the developer's liking, it can be modified. Once the developer is satisfied with the application, PPROG generates Paradox Applications Language (PAL) code. It creates procedures, functions, modules, and even libraries. The resultant scripts can be run under Paradox or the Paradox runtime program. Even after the code has been generated, PPROG can be used to modify the application if the PAL code has not been manually modified.

For my simple application, I spent an hour designing it and 90 minutes at the keyboard using PPROG, which then spent about 7 minutes generating more than 1,000 lines of PAL code. The program worked the first time. Writing and debugging the same PAL code would have taken me about two days, and that is being generous with a coding rate of 50 lines per hour.

As with Magic PC, PPROG has its limits. It cannot handle all situations and cases. Data entry, reports, and forms are somewhat more restricted compared with the possibilities under direct PAL control. Menus are limited to Paradox-style menus. However, the advantage of the approach is that a basic application can be generated quickly with PPROG and then manually fleshed out to incorporate more sophisticated tasks and actions.

The Data Entry Toolkit is a peculiar but powerful addition to Paradox. Previously, Paradox allowed the PAL programmer to display a form and wait for it to be filled in, returning control to the program when the programmer pressed one or more predefined keys. Now, however, something resembling a tiny object-oriented language is built into the wait facility and allows the programmer almost keystroke-by-keystroke control over the data-entry process. Although each action to be taken must be written procedurally, the set of actions for a given form are constructed in a more declarative style.

DECLARATIVE SPEED

Both Magic PC and PPROG do a lot of work in relatively short order and make few demands of developers other than that they know the basic product well (this is especially important for Paradox; novices can work with Magic PC) and approach the tool with the application design in hand.

Actually, the latter requirement is only partly necessary. Both tools support the development of an application in pieces and allow easy modification of a previously generated application, so both can be used by those who are trying to cobble a program together without any forethought. Obviously, the productivity advantage diminishes as the level of hacking increases.

For the developer with a well-designed application, preferably on paper, tools like these two result in an enormous productivity increase. The developer can avoid a large number of details, with the side effect of greatly reducing the opportunity for error.

Development speed and reliability are the two hallmarks of the declarative style of programming. What we need are more tools like Magic PC and PPROG that are even more sophisticated and resourceful so that developers can crank out applications of greater complexity and reliability.

NEW OLD FACES

A number of important changes are taking place at *PC Tech Journal*. A rash of promotions from within have given old faces new responsibilities, a reflection of the strong editorial staff that works so hard for you.

First, I am now the editorial director, a position that carries two mandates: to probe and report on products and trends and to lead *PC Tech Journal* into an ever more visible position in the industry and among systems developers and integrators. Working to see a magazine that I helped bring to life grow into a stronger voice is a role I relish.

I have turned the editorial helm over to Julie Anderson, who has been executive editor and my right hand for nearly four years. From the very first she showed the makings of a good editor: sharp mind, keen intuition, and the courage to ask why and say no—why are we interested in a topic; no, that article is not good enough for our pages. She is a computer scientist by training and a ten-year veteran of the computer industry. *PC Tech Journal* will grow and thrive under her able leadership.

Julie's first order of business was a staff reorganization. Two newly appointed executive editors will oversee our unique combination of journalistic excellence and technical accuracy: Susan Holly, our savvy chief copy editor for the last three years, will lead our writing/reporting efforts; Dave Methvin, our talented technical editor in charge of LAN coverage, will guide the technical side.

Not everyone is moving up the masthead. Our founding managing editor, Marjory Spraycar, is migrating to an entirely new masthead to be the editor of *Thinking Families*, a magazine she is launching for parents of elementary school children. She leaves with our very best wishes.

With superb organizational and editorial skills, Gail Shaffer glides into the managing editor's slot. Gail has been with us for three years, most recently serving as senior copy editor.

Bruce Ansley, a Texan with the magic touch when it comes to word processing and electronic typesetting procedures, assumes Gail's duties as senior copy editor.

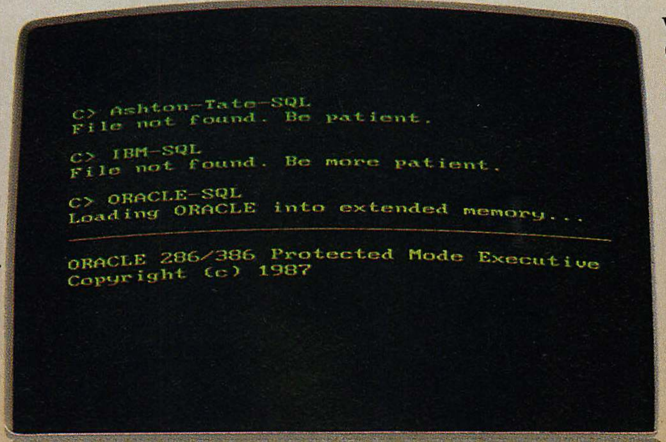
On the artistic side of the organization, Sharon Reuter, associate art director for 2½ years, will continue to lend us her visual imagination, now from the perspective of the art director's chair. Ina Saltz, a name you may remember from months past, is back on the masthead as creative director.

Trish Ledbetter, our administrative heart and soul, becomes our business manager, a position in which she will have increased responsibility for our business and financial affairs, as well as being an able assistant to both Julie and me.

I offer my heartiest congratulations to all for their dedication, hard work, and well-deserved promotions.

—*WF*

TRY ORACLE'S $1295 SQL DBMS FOR ONLY $199 TODAY. OR ORDER THEIR VAPORWARE.



If you're looking forward to OS/2 and the next generation of PC database management systems to enable you to build larger, higher-quality PC applications, you'll be interested to know that:

• Ashton-Tate has announced its intention to replace its outdated database technology with a SQL DBMS under OS/2, and

• IBM has announced its intention to offer SQL for OS/2 in its Extended Edition.

All you have to do is wait. Wait for OS/2 and SQL. Wait until mid-1988...or later! Meantime you keep developing with soon-to-be obsolete dBASE™ technology.

But you don't have to wait for SQL on the PC.

Oracle has it now. And you don't have to wait for OS/2 to run programs larger than 640K. The ORACLE® DBMS allows you to run OS/2-size programs under today's MS-DOS.™

ORACLE is the number-one selling DBMS on both minicomputers and mainframes. If you try PC ORACLE, we think you'll make us number-one on PCs, too. That's why we're making you this special offer.

We've sold thousands of copies of PC ORACLE for $1295. But now, for a limited time, we will send you the same, full-function version of ORACLE for only $199.*

That's right. Only $199 for a PC SQL DBMS that is identical to the ORACLE that runs on minicomputers and mainframes. $199 for the *only* PC DBMS that lets you write larger-than-640K, OS/2-size programs and run them under today's MS-DOS.

ORACLE makes it easy to write large, high-quality applications using industry-standard SQL. Applications that run faster. Applications that are easier to use and have more capability. Multi-user, networked applications. Applications that can be developed on MS-DOS today, and then run unchanged on OS/2, minicomputers and mainframes.

So stop writing dead-end dBASE code today. And stop writing cramped, limited-to-less-than-640K applications. Call 1-800-345-DBMS today. Or use the rightmost coupon. Or mail one of the other coupons. And wait.

ORACLE®

COMPATIBILITY • PORTABILITY • CONNECTABILITY

Call 1-800-345-DBMS today.

Dear Ashton-Tate,

20101 Hamilton Avenue
Torrance, CA 90502

Since you've announced your intentions to go to SQL, I really don't feel like developing obsolete applications in dBASE. I have every confidence your first attempt will be full-featured and bug-free. Send me what you've got, when you get it.

Name

Title

Company

Street

City

State Zip

Dear IBM,

Old Orchard Road
Armonk, NY 10504

Attached is a blank check. I keep reading that you've already announced that you're going to announce SQL for OS/2 and the Personal System/2.

When you do, fill in the check amount. Hope to hear from you sometime in the next couple years.

Name

Title

Company

Street

City

State Zip

Dear Oracle,

PC ORACLE • Oracle Corporation
20 Davis Drive • Belmont, CA 94002

I want to run OS/2 ORACLE on MS-DOS today. I agree to use this license only for application development. I understand all prices include applicable taxes and postage.

Send me *(check only one)*:

☐ PC ORACLE $199
☐ ORACLE + 1MB Extended memory $599
☐ ORACLE + 2MB Extended memory $799

I'm paying by ☐ Check ☐ VISA ☐ MC ☐ AMEX

Print Name Date

Company Title

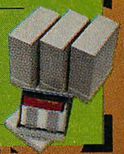
Street

City State Zip

Credit Card Number

Card Expiration Date

Signature PCTJ-9

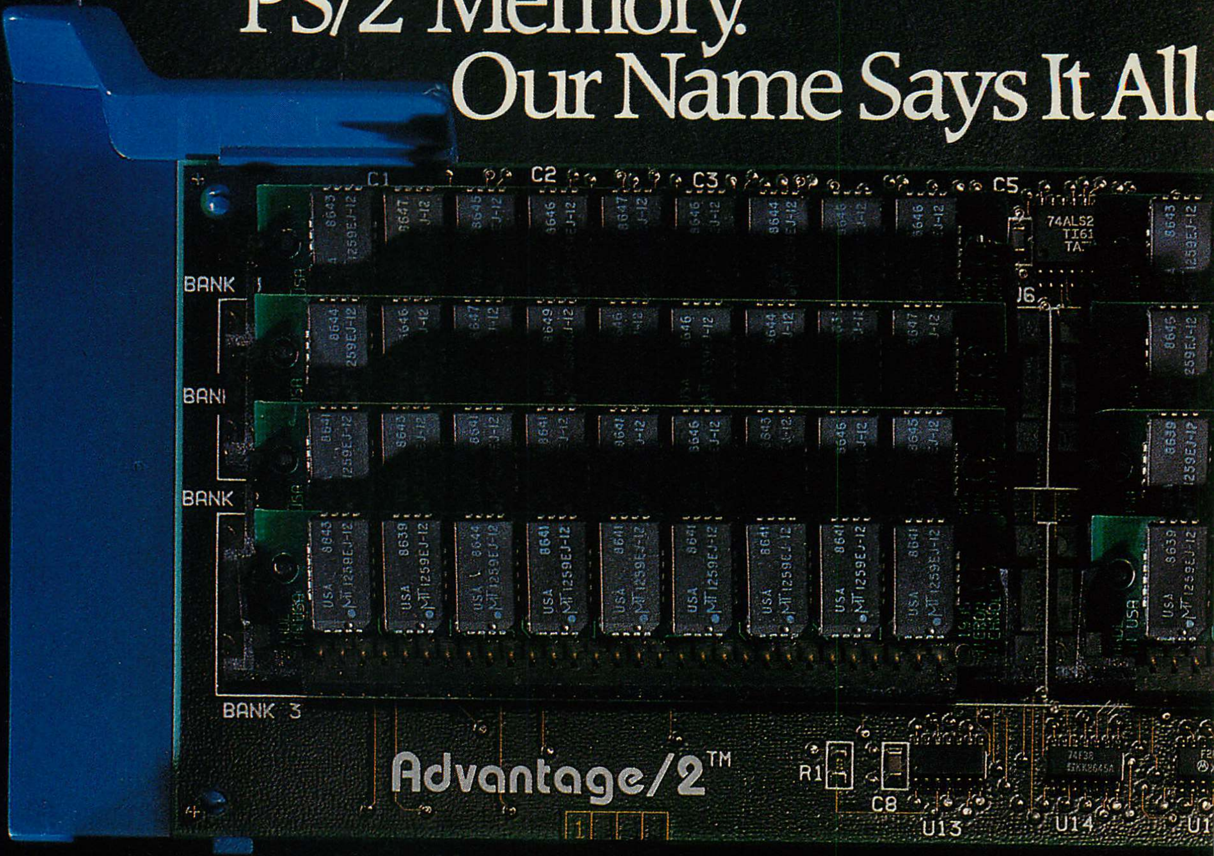


One Oracle Parkway • Belmont, CA 94002 • World Headquarters (415) 598-8000 • Calgary (403) 265-2622 • Ottawa (613) 238-2381 • Quebec (514) 337-0755 • Toronto (416) 596-7750 • ORACLE Systems Australia 61-2-959-5080 • ORACLE Europe 44-948-6911 • ORACLE Systems Hong Kong 852-5-266846

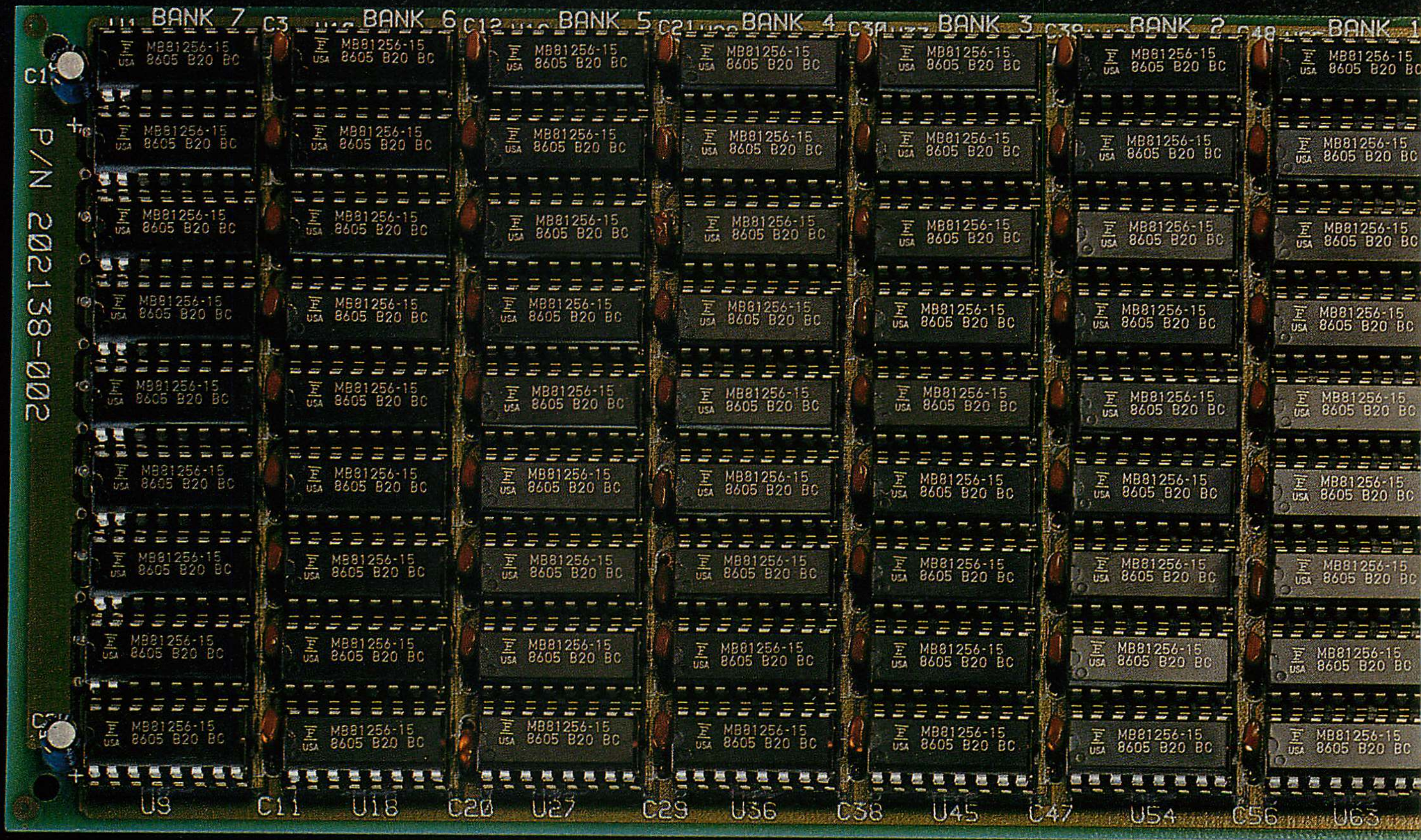
*PC ORACLE requires an "OS/2 ready" 286/386-based PC running DOS 3.1+ with 1MB extended memory (2MB is recommended) and a minimum 10MB hard disk drive.

TRBA

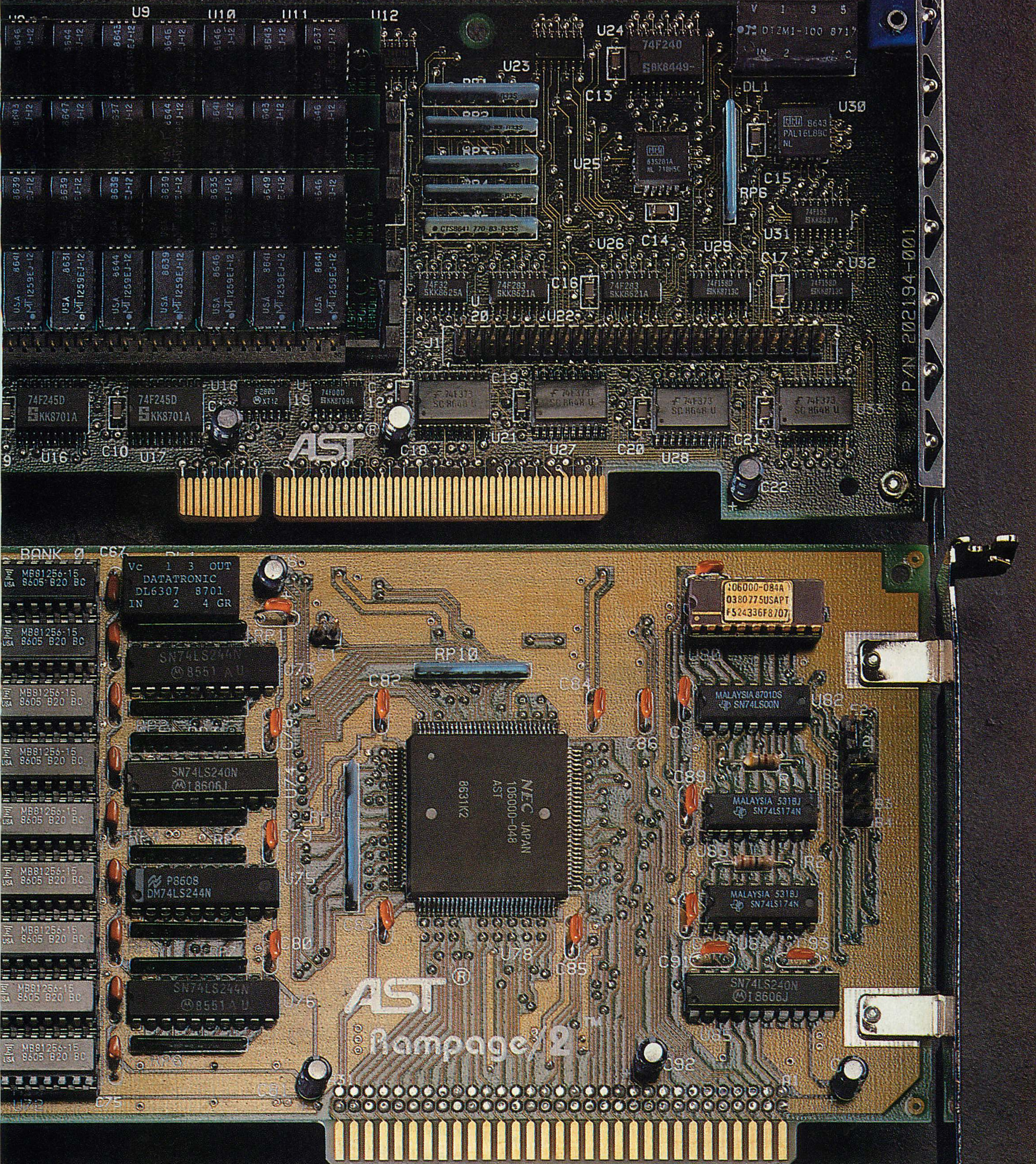
PS/2™ Memory. Our Name Says It All.



Advantage/2 for PS/2 50s and 60s offers up to 8 MB of memory, full Micro Channel™ support and optional I/O capabilities.



Rampage/2 for PS/2 30s provides up to 2 MB of Enhanced Expanded memory per board (8 MB per system), AST Smart Switch™ easy, switchless installation and SuperPak™ utility software.


AST
AST
Rampage/2™

EVEN MORE POWER AND FLEXIBILITY

BRIEF 2.0

Users and industry press alike have unanimously proclaimed BRIEF as the best program editor available today. Now, the best gets better, with the release of BRIEF 2.0.

Straight from the box, BRIEF offers an exceptional range of features. Many users find that BRIEF is the only editor they'll ever need, with features like real, multi-level Undo, flexible windowing and unlimited file size. But BRIEF has tremendous hidden power in its exclusive macro language. With it, you can turn BRIEF into your own custom editor containing the commands and features you desire. It's fast and easy.

Jerry Pournelle, columnist for BYTE magazine summed it all up by saying BRIEF is, **"Recommended. If you need a general purpose PC programming editor, look no further."** His point of view has been affirmed by rave reviews in C JOURNAL, COMPUTER LANGUAGE, DR. DOBB'S JOURNAL, DATA BASED ADVISOR, INFOWORLD AND PC MAGAZINE.

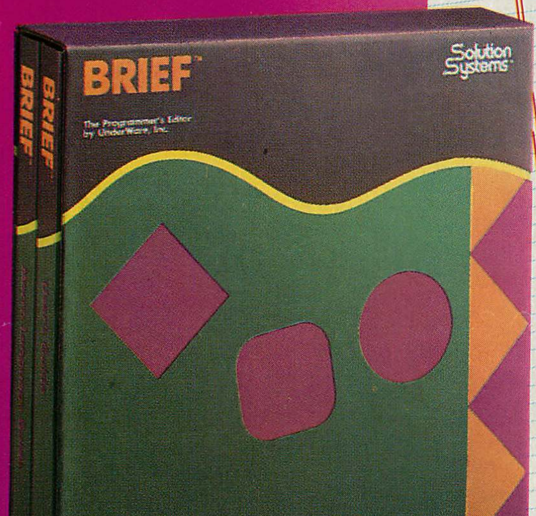
One user stated "BRIEF is one of the few pieces of software that I would dare call a masterpiece." Order BRIEF now and find out why. **BRIEF 2.0 is just $195. If you already own BRIEF, call for upgrade information.**

TO ORDER CALL: 1-800-821-2492 (in MA call 617-337-6963)

As always, BRIEF comes with a 30 day money-back satisfaction guarantee.

Solution Systems™

541 Main St.
Suite 410–P
South Weymouth, MA 02190
(617) 337-6963



Look at these BRIEF 2.0 enhancements!

Main Features:

- All new documentation with tutorials on basic editing, regular expressions **and** the BRIEF Macro Language.
- Setup program for easy installation and configuration. (Requires no knowledge of the macro language)
- Increased speed for sophisticated operations like Undo and Regular Expression Search.
- Expanded regular expressions, with matching over line boundaries.
- More block types, with marking by character, line or column.
- Command line editing (move cursor, add and delete characters, specify command parameters).
- Support for more programming languages.
- Optional borderless windows.
- Enhanced large display support, including wider displays.
- Reconfigurable indenting for C files (supports most indenting styles).

Plus the basic features that made BRIEF **SO** popular!

Basic Features:

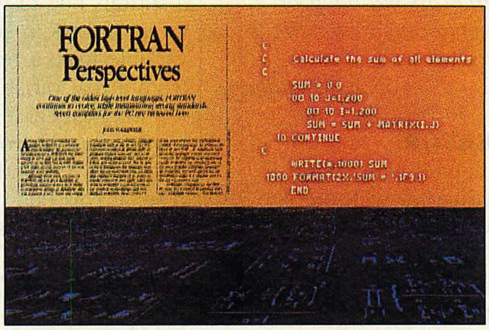
- Full multi-level Undo
- Windows
- Edit many files at once
- File size limited only by disk space
- Automatic language sensitive indentation

Requires an IBM PC or compatible with at least 192K RAM.

BRIEF is a trademark of UnderWare, Inc.

Solution Systems is a trademark of Solution Systems.

CIRCLE NO. 130 ON READER SERVICE CARD



MISSING LINKS

The article "Elegant Linkage" (Richard Halpern, May 1987, p. 132) on external subroutines for Turbo Pascal contains a number of errors in part because it generally agrees with the Turbo Pascal manual. This is a big mistake. The manual frequently differs with what Turbo Pascal actually does; at least, it does for all Turbo Pascal versions through 3.01A. The following corrections to the Turbo Pascal manual as well as to the article in May should be noted.

First, value parameters that are arrays or records are not passed as VAR parameters. The full array or record is copied to the stack just like any other value parameter. Second, value parameters that are sets do not require a full 32 bytes on the stack. Instead, Turbo uses the abbreviated format that it generally uses for storing sets. This format is described in the article's sidebar, "Data Representation in Turbo Pascal" (p. 134). Finally, external functions should not return a Boolean value by setting the Z flag because Turbo Pascal ignores this. Instead they must be returned in the AX register like any other scalar (0 = false, 1 = true).

In addition, the article omitted the following points, which should be noted by every programmer who writes external subroutines:

If a value parameter is a string, Turbo passes a field, the length of which equals the maximum defined length for the string. The first byte of the field contains the current length of the string and the subroutine should use only this length.

External procedures and functions must remove all parameters from the stack when they return because Turbo does not clear the stack. If this is not done, and the external subroutine is called repeatedly, garbage will be added to the stack by each iteration and the program ultimately will abort with a stack-heap collision error.

The stack is cleared by exiting with RET n where *n* equals the total number of bytes occupied by the passed parameters. Example one in figure 3 (p. 137) should have used a RET 8 (4 bytes for a string of maximum length 3, 4 bytes for a VAR parameter). Example three in figure 5 (p. 138) should have used a RET 6 (2 bytes for a scalar, 4 bytes for a VAR parameter).

When Turbo Pascal expects a function result to be returned in a register, it pointlessly reserves an area on the stack for the result. This area must be removed from the stack along with the parameters. For example, given the following definition:

```
FUNCTION JUNK(R1 : REAL) : Boolean;
EXTERNAL 'SUB.COM';
```

the machine code should put the Boolean result in AX and exit with a RET 7 (6 bytes for a real parameter and 1 byte for the Boolean result). Note that Turbo Pascal reserves only 1 byte on the stack for single-byte scalar results. In addition to being unnecessary, this is inconsistent with Turbo Pascal's practice of using a word to pass single-byte scalar parameters.

The three-instruction sequence on lines 2 through 4 of example one (figure 3) in the article is not the most efficient way to load registers DS and BX with the segment and offset of a VAR parameter. Instead, the single instruction LDS BX,[BP+12] should be used.

On the question of saving and restoring registers, the Turbo Pascal manual is somewhat confused, as Mr. Halpern points out. The programmer must save and restore BP, DS, or SS, but only if his code changes the contents of the register. To PUSH and POP any other registers is pointless because Turbo does not expect their contents to be preserved. To be sure, CS also must be preserved, but this happens automatically when a subroutine is called and then returns in the standard way.

Mr. Halpern implies that relocatability requirements for an external subroutine are the same as for a .COM module. Actually, they are more stringent for an external subroutine. Because a .COM module always will be loaded by DOS at offset 0100H, it need only be capable of running in any segment. An external subroutine, however, will be put at the next available offset in the code segment when its header is parsed by Turbo Pascal. This offset is unpredictable, so the code must be able to run at any offset. Therefore, a subroutine, unlike a .COM module, cannot make references to data at absolute offsets in the code segment.

Robert Pirko
New York, NY

I stand corrected on the question of Turbo Pascal manual errors. It may indeed be a big mistake to agree unquestioningly with the Turbo manual.

"Elegant Linkage" was meant to introduce some basic ideas that would encourage readers to do further experimentation on their own. Mr. Pirko's additions are valuable; however, some of them require comment.

His first point is incorrect. The length of the field is the maximum string length, plus one. In figure 1 (p. 135), this length is shown for the variable V4, a string of maximum length six, which occupies the seven bytes SP + 2 through SP + 8. The labeling on the right indicates clearly that the first byte is the length, but the 6 is probably a cause of confusion. In my string examples, I used strings with actual and maximum lengths that were the same. In that case, and only that case, would byte SP + 2 contain a 6.

The purpose of the code in example one (figure 3) is to illustrate the individual steps that are taking place, not to produce efficient code. As for saving and restoring registers, Mr. Pirko is correct. My recommendation

to save and restore all registers altered by the subroutine (an old habit of mine) should apply only to the BP, SS, and DS registers.

—Richard Halpern

FORTRAN FERVOR

Having experimented with Microsoft's FORTRAN 4.0 compiler with the supplied CodeView debugger (see "The State of FORTRAN," John Voglewede, June 1987, p. 92), I can attest that it is excellent and produces very fast code. Regrettably, in its current release it has a serious bug of which users should be aware: it errs in subtracting a complex variable from a real constant (but not from a real variable). For example, if C = (0,1), then 1 − C yields (1,1) rather than the correct value (1,−1). But, (1,0) − C and −C + 1 give correct results. This bug occurs in both single and double precision.

Edward Kausel
Massachusetts Institute of Technology
Cambridge, MA

Regarding your review of Microsoft FORTRAN 4.0, our experience differs considerably. When attempting to convert a large program (20,000 lines in 32 subroutines, several overlays) from MS FORTRAN 3.2 to 4.0, we finally had to give up in frustration.

We encountered problems with misstorage of results of calculations that could be solved only by preceding them with A = A, *A* being the variable to be calculated. Other problems had to do with the formatted read skipping blocks of records. Previously, all of this had worked in version 3.2. We also could not get a .LST file completed for some of the subroutines.

Microsoft told us they could not reproduce our first problem (note, we did not send Microsoft all of the source code), and that the file problem would be fixed at a later date.

Many of these problems no doubt are the result of working with large systems, but if bugs are present, when will they occur in smaller systems?

D. J. Meyer
Advanced Management Technologies
Houston, TX

Microsoft is in the process of eliminating several known bugs in its FORTRAN 4.0 compiler. A new revision (4.01) correcting these bugs is expected to be released shortly. Microsoft will provide this release at no cost to all FORTRAN 4.0 licensees.

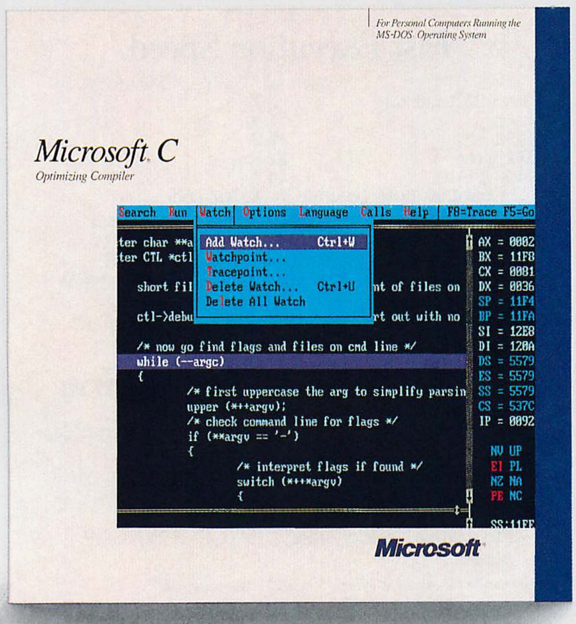
—John Voglewede

Thank you for recommending F77L to the serious PC FORTRAN programmer, in addition to recognizing our liberal license agreement and customer support programs. We do have some additional comments. The BIGARRAY bug found by Mr. Voglewede is fixed in version 2.21, which currently is shipping. The F77L compiler "fixes" itself each time it is used. This feature allows us to fix some user-reported problems the same day the bug is reported, then the current product is fixed so that new users will not have the problem, all without waiting for a new release. These fixes are available to all users on our bulletin board.

We take strong exception to Mr. Voglewede saying in the article that our compiler's "error-detection capabilities are only average." Lahey F77L users who download mainframe programs enthusiastically praise our compiler and runtime diagnostics. They report that F77L has diagnosed errors that have existed for several years.

Continued on p. 21

C5.0 has three features professional programmers can't live without.



Speed.

Fast Execution Speed.

	Microsoft® C 4.0	Microsoft C 5.0
Sieve (25 iterations)	5.7	3.3
Loop	11.0	0.0*
Float	19.9	0.1
Dhrystone	22.8	19.1
Pointer	14.2	7.4

- New optimizations generate the fastest code:
  - —Inline code generation. **NEW!**
  - —Loop optimizations: **NEW!**
    - —Loop invariant expression removal. **NEW!**
    - —Automatic register allocation of variables. **NEW!**
  - —Elimination of common sub expressions.
  - —Improved constant folding and value propagation.
- Fine tune your programs for even greater speed:
  - —Coding techniques for writing the fastest possible programs are included in the documentation. **NEW!**
  - —Segment Allocation Control:
    - —Group functions into the same segment to get faster NEAR calls. **NEW!**
    - —Specify which segments receive variables to yield faster NEAR references. **NEW!**
  - —Uses register variable declarations.
  - —Mix memory models using NEAR, FAR & HUGE pointers.

Benchmarks run on an IBM® Personal System/2.™ *Time is negligible.

Speed.

Fast Compilation.
Fast Prototyping.

Microsoft C Version 5.0 includes QuickC,™ which lets you edit, compile, debug, and execute in an integrated environment. It's ideal for prototyping.

- In-memory compilation at over 10,000 lines/minute. **NEW!**
- Built-in editor with parentheses, bracket and brace matching.
- Use the integrated debugger to animate through your program, add watch variables and set dynamic breakpoints. **NEW!**
- MAKE file is automatically generated for you. Simply indicate the modules you want to use, then MAKE recompiles and links only those modules that have changed. **NEW!**
- Full C 5.0 compatibility:
  - —Completely source and object code compatible.
  - —Emits CodeView®-supported executables.
  - —Identical compile/link command line switches.

For BIGERROR, I claim that all errors were reported. F77L accepts 31-character names and ignores blanks, so PROGRAM WITH LARGE ERRORS and REAL MATRIX SUM are correct. Minimally, 31-character names affect diagnosing BIGERROR lines 1, 4, 5, and 7. Our approach with F77L is precisely in keeping with the current draft of the FORTRAN-8x standard, which accepts names as long as 31-characters.

Finally, I would note that your article takes an approach that distinguishes it from the typical review offering only benchmark testing. LCS applauds *PC Tech Journal* for this policy.

Thomas M. Lahey, president
Lahey Computer Systems, Inc.
Incline Village, NV

During the past year and one-half, I have been porting (from the DEC VAX environment to the IBM PC family), maintaining, and modifying an 18,000-line FORTRAN program. I decided to use Lahey's F77L compiler for this effort because Microsoft FORTRAN 3.x would not handle all the features that were used in the program. Recently, I received a copy of Microsoft FORTRAN 4.0, and it was able to compile the program after a few minor revisions.

The Microsoft object and executable code was substantially larger than for Lahey's F77L. The program needs as much data space as possible. It is organized into overlays as is (Plink86 is used for linking). The compile time for Lahey's F77L 2.2 was about 1½ hours; Microsoft FORTRAN 4.0 took about 4¼ hours. Program execution speed was nearly identical for both.

Two things not mentioned in the article seem very important to me. First, the Lahey debugging package, SOLD, requires only about 20KB versus Microsoft CodeView's 200KB (which leaves me no data space). Second, Lahey's SOLD will find its way through overlays (I have used it to do this), whereas Microsoft has said that CodeView cannot be used with overlays.

Ronald E. Johnson
Northern Valley Software
Rancho Palos Verdes, CA

I have been using Lahey's F77L since the release of version 1.0 to develop a large, number-crunching system, but your article peaked my interest in the Microsoft compiler because your benchmark tests show the MS compiler comparing favorably with Lahey's. I conducted some extensive tests between the two using a commercial product that runs under MS FORTRAN 3.3. From start to finish, there is no question—in the workplace, Lahey's F77L beats Microsoft hands down.

Mark S. Gerber, Ph.D., president
M. S. Gerber & Associates, Inc.
Columbus, OH

In response to Mr. Lahey, the "average" error-detection capability attributed to the Lahey F77L was based on this compiler's performance relative to other FORTRAN compilers. The Lahey product did not detect syntax errors identified by others. F77L also failed to indicate that a compiler limit was exceeded during the compilation process.

While it is correct that blanks generally may be ignored in FORTRAN, Section 2.2 of the FORTRAN-77 standard limits the length of symbolic names to six characters or less. Proposed revisions to this standard notwithstanding, the lines in question should have been identified as containing syntax errors or extensions to the standard. (Also note that Lahey Computer Systems has informed PC Tech Journal *that an overlay linker and library file manager are available for use with F77L, as separate purchases.)*

Mr. Johnson is correct—the documentation supplied with Microsoft CodeView confirms that programs using overlays cannot be debugged.
—John Voglewede

"FORTRAN Perspectives" contains a misleading comparison of the accuracy of computations under the FORTRAN compilers that were being reviewed. PERFORM.F77, listed on page 110, and the accompanying expository description of the accuracy tests show that the figures returned bear no relation to the accuracy of computation.

I will illustrate the point with a simple example. Suppose that all computations carried out in the main loop of PERFORM.F77 are exact except one (a highly unlikely situation), and that the error returned for that one computation is, in terms of the program variables, C = 1.0. This might occur, for example, for the case A = 10,000, when computations are in single precision (such an error for this argument still would mean that about four of the available seven decimal places were correct in the computation of C). Then suppose D = 1.0 also. But the error statistic, the base-10 logarithm of D, is 0. If this result were included in figure 1 of the article, it would be recorded as 0 decimal digits of precision, a grossly incorrect interpretation.

Many other deficiencies can be noted: a rounding error for the case A = 10,000 contributing 100 million times as much to D as a rounding error for the case A = 1, accumulating the sum of squares of absolute errors instead of reporting the root mean square of the relative error, confusing mathematical exactness with computational exactness, and so on. But the point has been made. This benchmark is worthless in my opinion. If I were a manufacturer whose product had performed badly in your tests, I would be most unhappy. To publish such results is a disservice to your readers and to the manufacturers.

William J. Cody, Jr.
Glen Ellyn, IL

As pointed out by Mr. Cody, the results of the PERFORM.F77 program are accumulated calculational errors and do not represent the intrinsic precision of the compiler tested. Figure 1 was intended to demonstrate the relative trade-offs between speed and accuracy, and the differences between compilers using different options. The confusion between the accumulated error reported in this figure and the intrinsic precision of the compiler itself could be eliminated by Mr. Cody's suggestion (that the root mean square of the relative error be used rather than the sum of squares of the absolute errors). However, the relative comparisons of speed and accuracy shown in the figure remain valid regardless of whether this particular change is incorporated.
—John Voglewede

PC Tech Journal *would like to note additional corrections to "FORTRAN Perspectives." In describing figure 1 on page 94, column 3, the last sentence of the first full paragraph should read, "The precision of the result is simply the* negative *of the base-10 logarithm of the total accumulated error."*

Table 2 (p. 97) should show that Microsoft does not support in-line comments in its source code format and that Lahey's F77L CHARACTER length is reduced to 32,768. In table 3 (p. 98), the entire row of yes/no indicators for runtime distribution under License Agreement should be reversed to show that the DRI, Ellis, Lahey, and Microsoft compilers support it, while Prospero and Ryan-McFarland do not.
—JS

FORTRAN FERVOR, CONTINUED

Thank you for the command-line processor in "Command-Line Arguments for FORTRAN," (Programming Practices, John W. Ross, May 1987, p. 190). However, there appear to be two typographical errors in COMLIN.ASM on page 193. The correct values for the program prefix address should be

```
psp_seg equ 0000H
;program segment prefix segment
psp_seg equ 4F2H
;program segment prefix offset
```

After I made this correction, the program performed flawlessly with the Microsoft FORTRAN compiler 4.0.

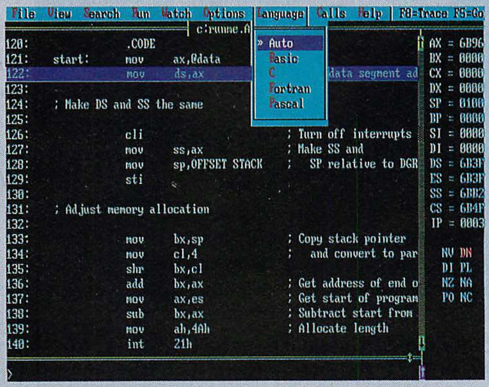
Werner Wothke, Ph.D.
Northwestern University
Evanston, IL

Although the address that is given by Dr. Wothke for the PSP is correct, the listing has no typographical error. There is no unique way of writing an

News about the Microsoft Language Family

Microsoft® Macro Assembler Version 5.0

Microsoft Macro Assembler Version 5.0 has a host of exciting new features that make assembly language programming easier and more powerful than ever! Microsoft Macro Assembler Version 5.0 now includes the Microsoft CodeView® window-oriented debugger and can assemble instructions written for the Intel® 80386 CPU. Comprehensive documentation and example programs help you write assembly code subroutines that can be called from other Microsoft languages such as C, BASIC, FORTRAN, and Pascal. Microsoft Macro Assembler Version 5.0 also assembles your programs 25 - 40% faster than Version 4.0.



Microsoft Macro Assembler's CodeView at Work.

Now it's easy to write assembly language subroutines for high-level language programs

If you have always wanted to write fast assembly language subroutines to make your programs fly, now is the time to try Microsoft Macro Assembler Version 5.0. Version 5.0 comes with a completely new Mixed-Language Programming Guide, which describes in clear, step-by-step language how to write, assemble, and link an assembly language subroutine with your Microsoft BASIC, C, FORTRAN, or Pascal programs. The examples from this manual are provided on disk so you can use them as templates, filling in your own code and leaving the prologue and epilogue instructions that make the interface function correctly.

Now Microsoft Macro Assembler Version 5.0 makes segmentation easy. Our new "simplified segmentation" directives are short, straightforward, and intuitive: To declare the correct model for your subroutine, just use the .MODEL directive and choose the model you need: SMALL, MEDIUM, COMPACT, LARGE, or HUGE. To start your data segment, just add a .DATA directive; to set your stack, add a .STACK directive; and to begin writing instructions, use the .CODE directive.

Latest DOS Versions:	
Microsoft C Compiler	4.00
Microsoft COBOL	2.20
Microsoft FORTRAN	4.01
Microsoft Macro Assembler	5.00
Microsoft Pascal	3.32
Microsoft QuickBASIC	3.00

Microsoft Macro Assembler now includes the CodeView source-level debugger

CodeView, Microsoft's famous debugger for its C and FORTRAN languages, now comes to the world of assembly language programming. Source-level debugging allows you to view your program just as you wrote it, including the comments and spacing that are especially important in helping you follow your program's execution. You can access variables by name, even if they aren't declared public, and see constants as names instead of as anonymous numbers.

Write programs for the 80386

Microsoft Macro Assembler Version 5.0 can assemble the new instructions available with the powerful 80386 CPU and the 80387 math coprocessor, as well as 80286 instructions that have been enhanced to work with 32-bit registers. You can also use the new 32-bit wide registers to write faster programs than ever by using such operations as 32-bit add and subtract and 32-bit multiply and divide without using multiple registers. For more powerful data access Microsoft Macro Assembler Version 5.0 allows you to choose any 32-bit register for indirect memory access, to use "scaling" for easy array access, and to move 32 bits of data directly from memory into a register and back.

For more information on the products and features discussed in the Newsletter, **write to: Microsoft** Languages Newsletter, 16011 NE 36th Way, Box 97017, Redmond, WA 98073-9717. **Or phone:** (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

8086 address in segment:offset format. His address, 0000:04F2, and the address published, 004F:0002, in fact, refer to the same location in memory.

—John W. Ross

In addition to Dr. Wothke's letter, several others were received that suggested worthy methods of retrieving command-line arguments in FORTRAN. The letters that follow are representative.

—JS

As Mr. Ross notes in his article, it is sometimes important to be able to retrieve command-line arguments from within a FORTRAN program—even though many PC-based FORTRAN compilers do not directly support such an activity. Occasionally, it is also useful to be able to read the DOS environment areas, which also may be accessed through the PSP.

Some FORTRAN compilers (notably, Lahey's F77L) provide FORTRAN functions to read these areas. Other major compilers (Ryan-McFarland and Microsoft, for example) do not. We have been using both RM FORTRAN and MS FORTRAN to port large statistical systems from mainframes to PCs. We had to overcome the same problem that Mr. Ross faced. I think we ended up with a simpler solution.

DOS 3.x includes an interrupt (62H, get PSP address) that retrieves the PSP address from any point in a program. All DOS versions from 2.0 on have an undocumented DOS interrupt (51H) that achieves the same result (it appears to work in a manner identical to the fully documented 62H version). The use of these interrupts obviates the necessity of Mr. Ross's "header" program, and greatly simplifies the production and linkage of the routines involved. Obviously, one is running some risk with versions of DOS earlier than 3.0, as the interrupt is undocumented and may change. For the time being, however, it works.

Brent C. James, M.D.
Intermountain Health Care, Inc.
Salt Lake City, UT

Microsoft FORTRAN does have a limited built-in capacity to use command-line arguments. If you want to pass file names through the argument list, they can be opened by leaving the file name blank on the OPEN statement, that is, OPEN(1,FILE = ' ') to open a file to unit 1. The first OPEN references the first argument, the second OPEN references the second argument, and so on.

The file names can be retrieved by the program using the INQUIRE statement.

Microsoft FORTRAN 4.0 requires assembly language subroutines to save the DI and SI registers, which was not required in previous versions. This will require a minor modification in the COMLIN.ASM program (as well as most other assembly language subroutine written for previous FORTRAN versions). I missed the blank file name trick several times before I found it in the manuals. Now I use it frequently.

James A. Parsly
Knoxville, TN

PREMIUM FOOTNOTES

In subsequent testing of the AST Premium/286 reviewed in the June 1987 issue (see "Compatibility and Performance: Premium/286, Steven Armbrust and Ted Forgeron, p. 74), it was discovered that the IBM Token-Ring Adapter will not operate when used with the IBM TOKREUI.COM driver supplied with the adapter. As of April 2, IBM has replaced TOKREUI with the LAN Support Program, which does function correctly on the Premium.

Incompatibility problems of this type on the Premium also can be solved by removing jumper E2 on the AST FastRAM board; this adds a wait state to memory accesses. However, adding a wait state by removing this jumper will negate much of the Premium's performance advantage. The jumper is documented in the revision B manual (the revision letter is located on the bottom left-hand corner of the manual's back cover).

—*DM*

ERRATA

In the "Graphics Intelligence" article (Ed McNierney, July 1987, p. 46), figures 4 and 5 are switched.

COMMENT AT WILL

All letters to the editor should be directed to Editor, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. Correspondence also can be submitted over MCI Mail to PCTECH.

Although *PC Tech Journal* cannot publish all letters received, every effort is made to answer as many as possible. Please keep letters brief and to the point, and include name, mailing address, and telephone number; when a letter is lengthy, a diskette is appreciated.

WINDOWS FOR DATA™

The first choice of professional C programmers

"Windows for Data is the best programming tool I've ever used.
It's the most flexible I've seen.
Whenever I've wanted to do something,
I've been able to find a way."

Steven Weiss,
Stratford Systems

Professionals choose our tools because they are designed, crafted, and supported for professionals. Here at Vermont Creative Software, we understand that performance and pleasure in programming derive from more than a long list of functions. **Windows for Data** provides:

PROFESSIONAL FLEXIBILITY: Our customers repeatedly tell us how they've used WFD in ways we never imagined - but which we anticipated by designing WFD for unprecedented adaptability. Virtually every capability and feature can be modified to meet special needs. You will be amazed at what you can do with WFD.

PROFESSIONAL PERFORMANCE: Screen output is crisp and fast. Windows, menus, and data-entry forms snap up and down from the screen. WFD is built upon and includes **Windows for C, the windowing system rated #1 in speed and overall quality** in PC Tech Journal (William Hunt, July 1985).

PROFESSIONAL RELIABILITY: An unreliable tool is worse than no tool at all. VCS products are known in the industry for their exceptional reliability. Ask anyone who owns one.

PROFESSIONAL DOCUMENTATION: Over 600 pages of documentation provide step-by-step explanations for each major application, a reference page for each function, listings of functions alphabetically and by usage, and a fully cross-referenced index. Extensive tutorials and demonstration programs assist learning.

PROFESSIONAL TECHNICAL SUPPORT: The same expert programmers that develop our products provide prompt, knowledgeable technical support.

PROFESSIONAL PORTABILITY: High-performance versions of VCS products are available for XENIX, UNIX, and VMS, as well as DOS. No royalties on end-user applications.

OUR CHALLENGE AND GUARANTEE

If you have an application where no other tool can do the job, try **Windows for Data.** If it doesn't help you solve your problem, RETURN FOR A FULL REFUND. YOU MUST BE SATISFIED.

Ask for **FREE DEMO DISKETTE**

Vermont Creative Software

21 Elm Ave.
Richford, VT 05476
Telex: 510-601-4160 VCSOFT
Tel.: 802-848-7731

Prices: PCDOS* $395; XENIX, VMS, UNIX Call.
*PCDOS specify C compiler.

WINDOWS FOR DATA

for DOS, UNIX, VMS ...

The complete windowing data entry, menu, and help system that does the hard job others can't — we **guarantee** it!

Pop-up data entry windows; field types for all C data types, plus decimals, dates, and times; auto conversion to and from strings for all field types; system and user supplied validation functions; range checking; required, must-fill, and protected fields; free-form movement; multiple-choice field entry; scrollable sub-forms. Branch and nest windows, forms, and menus.

Complete context-sentitive help system with pop-up windows and scrollable text.

Pop-up, pull-down, scrollable, and Lotus-style menus.

NEW FOR DEBUGGING: Exclusive **VCS Error Traceback System** automatically identifies the location and cause of program errors. Eliminates the need to code error checks on all function calls! **VCS Memory Integrity Checking** helps catch those hard-to-detect, memory-corruption errors.

NEW FOR ERROR HANDLING: Install your own error handler to be called whenever a function detects an error.

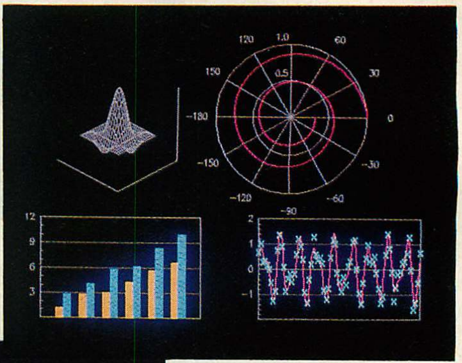
NEW FORM LAYOUT UTILITY simplifies form design.

SPEED-UP YOUR PC GRAPHICS!

With the GSS® Graphics Development Toolkit and your favorite language,* you can write ***faster*** graphics applications in ***record time.***

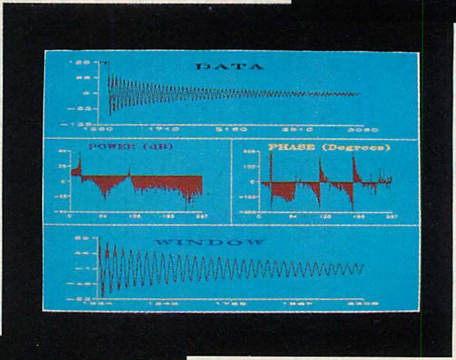
Increase Your Application Performance

GSS drivers are written directly to the hardware, so your applications run fast. And the latest display cards are supported, including high-performance boards from Video-7, Quadram, Emulex, and others. Low memory overhead also promotes faster performance. The GSS software typically requires only 75Kb at run-time.



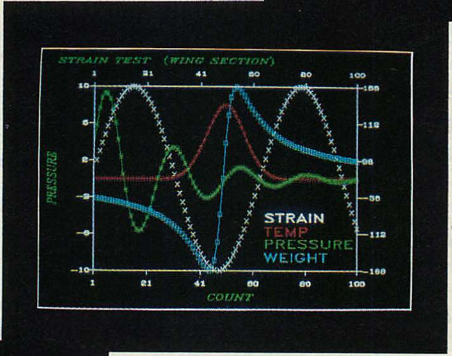
Reduce Your Time To Market

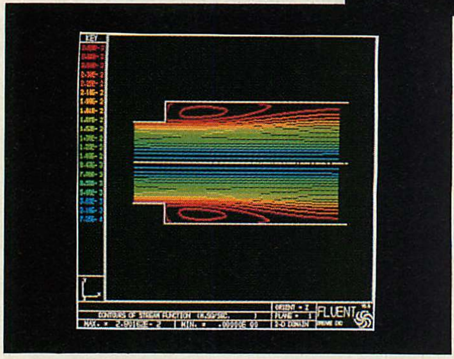
GSS provides the graphics functions and device drivers, so you spend less time in development. Call over 100 high-level functions for lines, arcs, text, bitmaps, clipping, and more. Support numerous mice, displays, printers and plotters, including EGA displays, HP plotters, Epson LQ printers and the HP LaserJet+.



Build The Future Into Your Software

Your applications will work with new graphics devices as new GSS drivers become available. You won't have to modify your program! And when you use the Graphics Development Toolkit, your graphics software will be source code compatible with the new features of future DOS versions.





Our complete GSS Line:

	List:	PC Brand:
CGI Dvlpmt Toolkit	$495	$375
Kernel System	$495	$375
Kernel for IBM RT	$795	$645
Metafile Interpreter	$295	$235

For an in-depth look at creating high-performance PC AT graphics, attend the nearest GSS Seminar:

San Jose	June 11
Washington, D.C.	June 18
Los Angeles	July 9
Boston	July 23
San Jose	Aug. 6
New York City	Aug. 20
Orange County, CA	Sep. 10
Boston	Sep. 24
San Jose	Oct. 8
Atlanta	Oct. 22

Call PC BRAND to register now! Registration fee, $25.

***languages supported:**

Lattice C; Microsoft C; Lahey FORTRAN; Microsoft FORTRAN; R/M FORTRAN; IBM FORTRAN; IBM Professional FORTRAN; Microsoft Pascal; IBM Pascal; Microsoft BASIC Compiler; Microsoft Quick BASIC; IBM BASIC Compiler; Microsoft Macro Assembler; IBM Macro Assembler.

Photos provided by CREARE, INTEGRATED SYSTEMS, and OPTIM ELECTRONICS

To Order Your GSS Software Today Call

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

PC BRAND™: CAREFULLY CHOSEN PROGRAMMER TOOLS

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out programs themselves** if product code begins with E, T, or L through N — even if it means breaking the disk seal. Some developers do pose limits, so for products beginning with other letters, opening sealed disks constitutes acceptance. But you can at least review the manual. There's just nothing stopping your buying from PC Brand.

BRIEF *Version 2.0 Extends Lead* NEW!

Brief has come forth with version 2.0 offering editing templates for added languages, faster undo and search with matching over multiple lines, new types of block marking, large screen support, new documentation, and tutorials.

This runaway success has eclipsed all other editors by sporting every feature you've ever contemplated. Text, from keyboard or files, is housed in multiple buffers, and scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers or deletion, with as many backward "undos" as you want.

With a growing list of C compilers, you can compile without ever leaving Brief. It grabs the compiler's report of problem lines, and marches you through your source code for repairs.

Parts of Brief were written with its own Lisp-like macro language which has structure, 32-character variable names, conditional execution, loops, and even readability. Nothing like the hieroglyphs we've seen elsewhere. Bulletin board and public domain disks with macros. "One of the best investments you can make," *PC Magazine*. To top it off, there's a 30-day money-back trial period. List: **$195**, Us: **Call.**

CLIPPER *From Nantucket*

Compile dBASE for Speed, Protection

Clipper™ turns lumbering dBASE® into a speed dMON with benefits bobbing in its wake: your source code is submerged from public view, you can distribute your compiled application without royalties, and your customers don't even need copies of dBASE! The Spring '87 Clipper offers index files compatible with dBase III Plus, and networking capabilities to run compiled programs on major networks supporting DOS 3.1 with no restrictions on number of users. Clipper offers arrays, menu-building commands, user-defined functions, context-sensitive help techniques for applications, a debugger, and it supports Expanded Memory. It goes well beyond dBASE with 1,024 fields per data base and 2,048 active memory variables.

Clipper has the power to save and restore multiple screens to and from memory variables. You can also create overlays, call object modules compiled in other languages, and create function libraries to link with your applications. Power and flexibility make it the #1 dBASE compiler. List: **$695**, PC Brand: **Call.**

McMAX *From Nantucket* NEW!

Like dBASE for the Macintosh

McMax™ is like running dBASE on the Macintosh. It combines an easy-to-use menu-driven ASSIST mode using the Mac interface, an interactive command mode like dBASE at the dot prompt, and an application programming language fully compatible with dBase III. It gives you the power to create dBASE language applications on the Macintosh and transfer back and forth to the IBM® world. McMax accommodates up to 16 million records, 32,000 characters per record, 255 characters per field, and up to 32 files open concurrently. No copy protection. List: **$295**, PC Brand: **Call**

MICROSOFT C 4.0

A Great C Battle Rages and You're Winning

It bundles a source debugger and a "make", and sports a "huge" memory model permitting single data objects larger than 64k, but what's really impressive about Microsoft C are the benchmarks reported in Dr. Dobb's. Microsoft runs away from a field of 17 winning 11 of 27 benchmarks.

The CodeView™ debugger uses windows to show everything on one screen: source alongside disassembled object, variables, stack and registers. Drop down windows obviate learning of commands. "A source-level debugger that puts the rest to shame" said Dobb's.

Microsoft C has five memory models for code and data, plus non-library support for another thirteen, and boasts alternate math packages for speed versus accuracy, with or without 8087/80287 chips.

Both linker and library manager are part of the package, as is the "make", which knows how to rebuild any size project by compiling only elements which have changed.

It is reportedly used by Lotus, Ashton-Tate and, fittingly, Microsoft itself to develop Windows. Dobb's calls it "the best MS-DOS C development environment value today [for] virtually any kind of program conceivable." 320k suggested.

Ask for:	List:	PC Brand:
G0500	**$450**	**$295**

HALO GRAPHICS SYSTEM *Multi-Board Graphics Library*

The premier graphics library that got the ball rolling for PC-based graphics and has grown so omnipotent that it supports over 25 graphics boards — including IBM's EGA and Nr. 9 Revolution's hi-res series — and has a multitude of mouse and printer drivers. All that in each box. Separate C versions for Lattice, M'soft, Aztec, CI86. What does Multi-Halo do? A down to the last pixel graphics library plus functions to reset drivers so distributed program can run on anything. Wonderful value for single license. Costly royalties though for redistribution. Specify: S0315 & Language. List: **$300.** We: **$219.** With Dr. Halo II, a free-standing "paint": List: **$440,** Us: **$299.**

C-TREE & R-TREE *B-Tree File Manager Now Has Report Generator*

c-tree: Seasoned to perfection, c-tree™ is the only major b-tree file manager with network support in the standard low-cost version. It allows multiple users to access an index file simultaneously even during updating. Record-locking routines are provided for DOS 3.x, UNIX and XENIX.

C-tree even comes in C source code, yet there are no royalties. Source sticks to K&R, so C-tree is portable. Tests in many environments prove it.

C-tree permits any number of keys for a data file—alpha, numeric, even floating point. It handles files with varied record lengths and keeps multiple keys in one index file. Has both high level ISAM routines to handle details with minimum coding, and decomposed step-by-step functions you can access directly. In short, you get the works.

r-tree: thousands of c-tree users (and you) now have a suddenly expanded ability to produce ad hoc reports from files maintained by c-tree (v. 4.1 and up). Just link a file description to the r-tree library, and all you need is an ordinary text editor to write any number of report scripts with no further C coding. Reports can access data in several files, select on criteria, join the findings into new logical records, sort them, calculate new fields and columns, tabulate by any number of control breaks. The script files show a visual representation of the report image for easy creation and maintenance of even elaborate layouts. r-tree™ comes in source, boasts the same portability as c-tree, and fits any compiler.

	List:	PC Brand:
c-tree: F0660	**$395**	**$329**
r-tree: F0665	**$295**	**$245**
Combined:	**$650**	**$541**

dBC *Lattice Library Maintains dBASE Compatible Files With the Power and Speed of C*

dBC™ links C to dBASE. It creates and maintains files and their indexes which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dBC can use any files created by dBASE. Now C and dBASE can operate on the same data bases interchangeably.

That opens up the widespread culture of dBASE installations to exploitation by C programmers. Tap that market, avoid the resident dBASE language, and gain the advantages of C with this single product.

dBC's functions parallel all dBASE's file handling commands, many decomposed to give closer control. Each backed by demo source files on disk.

Use dBC for custom work for clients, or on its own. It's a complete ISAM file manager for C whether or not dBASE will be used in tandem, supports all four memory models, and can have sixteen index and data files open. Big discount to buyers of both dBASE II and III versions. Specify Lattice, Microsoft 3.x, or DeSmet.

Versions:		List:	PC Brand:
L00II	For dBASE II	**$250**	**$195**
LCCII	With Source	**$500**	**$390**
L0III	For dBASE III	**$250**	**$195**
LCIII	With Source	**$500**	**$390**

POLYTRON VERSION CONTROL

Source Code Control for Any Language

PVCS allows programmers, project managers, librarians and system administrators to control the proliferation of revisions and versions of source code in software systems. Independent programmers, the leading software publishers and LAN companies, and hundreds of Fortune 1000 companies rely on PVCS to store and retrieve multiple revisions of text. It maintains a complete history of revisions as an "audit trail", generates status reports, and uses intelligent "difference detection" to minimize disk space for each new version.

On Corporate and Network PVCS simultaneous changes to a module are merged into a single new version. If changes conflict, the user is notified. The "Logfiles" used to track changes are interchangeable between any PVCS product.

Corporate PVCS is for multiple programmers. It includes "branching" to maintain code when programs evolve on multiple paths. Personal PVCS offers most of the power and flexibility of corporate PVCS, but excludes multiple programmer features. Network PVCS is the Corporate version for LANs. File locking and security levels can be tailored to each project.

PolyMake Still using a prehistoric Make? Step up to PolyMake, the most popular and powerful Make utility. Most flexible macros of any Make. Now integrated with PVCS and Poly-Librarian for faster performance and precise configuration management.

Ask for:	List	PC Brand:
Personal PVCS	**$149**	**$109**
Corporate PVCS	**$395**	**$309**
Network PVCS	**Call**	**Call**
PolyMake	**$149**	**$109**

WHY US?:

Latest versions of all products • Everything listed is in stock • Shipped 24 hours or sooner • No surcharge for credit card or COD purchase. **NEED TERMS?** On-the-spot credit to most public companies, government, educational, medical institutions. **LOOKING FOR SOMETHING?** We can get many more products — just ask! **NEED MORE INFO?** Our Catalog and literature cover just about everything.

For Orders or Literature, Call Us at....

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

TODAY's TOP QUALITY AIDS TO PROGRAMMING PRODUCTIVITY

SUPER SOURCE

Two great debugging and documentation tools that support C, Modula, Pascal, BASIC, and dBASE:

Source Print

Author Alan Simpson writes "the best overall debugging technique is to... draw lines to connect all the IFs and DO WHILEs etc. with their ENDIFs and ENDDOs. Use a pen or pencil to..." Well thanks, Alan, but we'll let Source Print do it for us. It draws vertical lines to connect all such beginnings and endings in a vivid display of your program's organization. And it can print programs with page numbers, headings, line numbers, indent automatically, throw in a table of contents and a cross-reference index. About Source Print *PC Magazine* said, "Occasionally a utility comes along that makes a programmer's life much easier." List: **$75**, Ours: **$60**.

..and Tree Diagrammer

Tree Diagrammer prints an organization chart of your program's structure showing the hierarchy of function, procedure, and subroutine calls. Shows at a glance what routines call each other for clearer debugging. Every shop should have this important documentation tool. List: **$55**, Ours: **$45**.

PANEL PLUS *Library Source Code Gives It Complete Portability*

NEW! NO ROYALTIES!

There are no end of tools for screen design and data entry, but none quite like Panel Plus. Design a screen under program control, use Panel's utility to "run" and test it field by field, then pass it to Panel's code generator which delivers C source code. Options style the code to your compiler's liking, and you can of course do what you like to the source afterward. The code calls Panel Plus's function library, but now *the library comes in source*, so everything produced is highly portable. Not like other screen managers delivered as object libraries and which leave *you* to write the detailed code.

Panel Plus will operate in graphics mode via interfaces to graphics products it supports and can utilize the EGA's 43-line screen. Low-level I/O functions adapt it to various keyboards, screens, operating systems.

Panel's newest incarnation has every imaginable feature. A single screen design can have 1000 fields stacked as visual overlays up to 127 levels deep or as pop-ups. Groups of fields can be moved between levels. Screens can be output as compilable code or stored on disk for loading at run-time. Each field can be boxed, colored, multi-row, word-wrapped, and scrolled horizontally and vertically if larger than its on-screen view aperture. It can be assigned its own help and error message, can be told to accept certain characters, or to match a picture, and to check data after entry—proper dates, number ranges, etc.—using Panel's or your own validation routines. You can add your routines to Panel's test utility because even it comes as source. Fields are accessed in any order and control reverts to your application program after each field for choice of action.

For past Panelists, the new version has smaller and faster field and screen functions, tighter granularity, and an enhanced, reworked library. Major tool for the serious developer. List: **$495**, PC Brand: **$395**.

Shopping List for the Power Workbench

	LIST	US
ARITY PRODUCTS		
Arity Combination Package	1095	979
Expert System Development Pkg	295	229
File Interchange Toolkit	50	44
PROLOG Compiler & Interpreter	650	569
Screen Design Toolkit	50	44
SQL Development Package	295	229
Arity PROLOG Interpreter	295	229
Arity Standard Prolog	95	77
AI-EXPERT SYSTEMS		
Autointelligence *by IntelligenceWare*	990	739
Experteach II *by IntelligenceWare*	475	339
EXSYS *Development Software by EXSYS*	395	309
EXSYS Runtime System	600	469
Insight 2+ *by Level Five Research*	485	379
Intelligence/Compiler *IntelligenceWare*	990	739
AI-LISP LANGUAGE		
IQCLISP *by Integral Quality*	300	Call
IQLISP *by Integral Quality*	270	Call
Microsoft LISP *Common LISP*	250	189
TransLISP *from Solution Systems*	95	Call
TransLISP PLUS *from Solution Systems*	195	Call
AI-PROLOG LANGUAGE		
APT *Active Prolog Tutor from Solution Sys.*	65	Call
MPROLOG Language Primer *LOGICWARE*	50	45
MPROLOG P500 *by LOGICWARE*	495	395
MPROLOG P550 *by LOGICWARE*	220	175
Prolog-86 *from Solution Systems*	125	Call
Prolog-86 Plus *from Solution Systems*	250	Call
Turbo PROLOG *by Borland Int'l*	100	63
Turbo PROLOG Toolbox *by Borland Int'l*	100	64
ASSEMBLERS & DEBUGGERS		
Advanced Trace-86 *Morgan Debugger*	175	119
C-Sprite Debugger *by Lattice, source level*	175	139
Microsoft Macro Assembler *with Utilities*	150	109
PASM86 *by Phoenix, Macro Assembler*	195	125
Periscope I Debugger...*The Periscope Co.*	345	289
Periscope II *with NMI Breakout Switch*	175	139
Periscope II-X *software only*	145	105
Periscope III *with Advanced Board...New*	995	Call
BASIC LANGUAGE		
Microsoft BASIC Interpreter *for XENIX*	350	295
Microsoft QuickBASIC Compiler...*3.0*	99	79
BORLAND PRODUCTS		
Reflex & Reflex Workshop	200	129
Reflex *Data Base System*	150	89
Reflex Workshop	70	45
Turbo Basic...*New*	100	64
Turbo C Compiler...*New*	100	64
Turbo Lightning	100	64
Turbo Pascal & Tutor...*New*	125	85
Turbo Pascal with 8087 & BCD	100	64
Turbo PROLOG Compiler	100	63
Turbo PROLOG Toolbox	100	64
Turbo Tutor	40	28
C COMPILERS		
C86 PLUS *by Computer Innovations...New*	497	397
Lattice C Compiler *from Lattice*	500	299
Let's C Compiler *by Mark Williams*	75	55
with CSD Source Level Debugger	125	90
MWC-86: Mark Williams C Development	495	369
Microsoft C Compiler *...4.0*	450	295
Turbo C Compiler *by Borland...New*	100	64
C INTERPRETERS		
C-Terp *by Gimpel Software*	300	249
Instant C *by Rational Systems*	500	395
Interactive-C *by IMPACC with debugging*	249	219
RUN/C Professional *from Lifeboat*	250	185
COMMUNICATIONS		
Asynch Manager *by Blaise, for C or Pascal*	175	117

	LIST	US
Greenleaf Communications *by Greenleaf*	185	139
C UTILITY LIBRARIES		
Blaise C Tools Plus	175	117
Blaise C Tools	125	84
Blaise C Tools 2	100	67
C Food Smorgasbord *by Lattice*	150	109
C Utility Library *by Essential, 300 functions*	185	139
Greenleaf Functions *by Greenleaf Software*	185	139
COBOL LANGUAGE		
RM/COBOL...*see Ryan-McFarland Prod.*		
Microsoft COBOL Compiler	700	499
Microsoft COBOL Compiler *for XENIX*	995	795
Micro Focus COBOL...*see Micro Focus Prod.*		
dBASE SUPPORT		
BRIEF/dBRIEF...*Brief for DBASE III*	275	Call
CLIPPER...*from Nantucket*	695	Call
McMAX...*dBASE for MAC from Nantucket*	295	Call
dBx..*dBASE to C Translator by DESKTOP AI*	350	299
with Library Source Code	550	469
with Full Source Code	1000	855
dBC...*from Lattice...maintains DBASE files*	250	195
with source	500	390
dBC III Plus...*supports multi-user DBASE*	750	595
with source	1500	1185
DEVELOPMENT TOOLS		
BASTOC *by JMI, convert BASIC to C*	495	399
BASIC-C *BASIC's functions added to C*	175	139
Report Option *Softcraft Btrieve Report Gen.*	145	128
Xtrieve *Softcraft Query Utility for Btrieve*	245	220
Code Sifter *Profiler by David Smith*	119	89
Dan Bricklin's Demo Program *Prototyper*	75	69
LMK *from Lattice, "make" like UNIX*	195	149
Microsoft Window Development Toolkit	500	365
ESSENTIAL PRODUCTS		
C Utility Library	185	119
Essential Graphics	250	183
Essential Comm Library	185	125
with Breakout Debugger	250	189
FILE MANAGERS		
Btrieve *by Softcraft, no royalties*	250	195
Btrieve Network *by Softcraft*	595	465
C-Tree *by FairCom - no royalties, source*	395	329
R-Tree *by FairCom-Report Generator*	295	245
C-Tree & R-Tree Combo *by FairCom*	650	541
dbVista *single user DBMS by Raima*	195	139
dbVista *multi-user DBMS*	495	399
Opt-Tech Sort *Can sort Btrieve files*	149	105
FORTRAN LANGUAGE		
RM/FORTRAN *by Ryan McFarland*	595	499
FORTRAN Libraries...*by Alpha Computer*	70	45
Microsoft FORTRAN *Links w/Microsoft C*	450	281
Microsoft FORTRAN *for XENIX*	695	546
Scientific Subroutine Package *by Alpha*	295	239
GRAPHICS		
Essential Graphics *Essential, no royalties*	250	210
GSS Graphics Development Toolkit	495	375
GSS Kernel System *by Graphic Software*	495	375
GSS Metafile Interpreter	295	235
Halo *by Media Cybernetics*	300	219
with Dr. Halo II	440	299
Halo for Microsoft *includes all fonts*	595	434
MICRO FOCUS PRODUCTS		
Micro Focus Level II COBOL *w/Animator*	495	395
Level II COBOL	349	279
Level II Animator	195	155
Micro Focus Personal COBOL	149	99
Forms-2	295	235
MODULA-2 LANGUAGE		
MODULA-2 Apprentice Pkg *by LOGITECH*	99	79

	LIST	US
MODULA-2 Magic Pkg *by LOGITECH*	99	79
MODULA-2 ROM Pkg & Cross RT Debugger	299	239
MODULA-2 Window Pkg *by LOGITECH*	49	39
MODULA-2 Wizard's Pkg *by LOGITECH*	199	159
OTHER LANGUAGES & UTILITIES		
Source Print...*source code formatter*	75	60
Tree Diagrammer..*source code diagrammer*	55	45
Help/Control *by MDS...123 style Help*	125	109
Microsoft MuMath *includes MuSimp*	300	199
Microsoft Pascal Compiler *Links w/M'soft C.*	300	199
Microsoft Pascal Compiler *for XENIX*	695	546
Turbo Power Tools+ *by Blaise*	100	67
Advantage C++ *by Lifeboat Assoc.*	495	435
PHOENIX PRODUCTS		
PASM86 *Macro Assembler*	195	125
PDisk *Phoenix's new disk manager*	195	125
Pfantasy Pac *six products...New Price*	995	Call
PFinish *EXE performance analyzer*	395	235
Pfix86 Plus *Symbolic Debugger*	395	235
PforCe *vast library*	395	235
PforCe++...*Function Library for C++*	395	235
Plink86 Plus *Utilizes memory for overlays*	495	325
Pmaker *like UNIX "make"*	125	85
Pmate *with Macros*	195	115
Pre-C *UNIX "lint"-alike*	295	174
PTel *Binary File Communicator*	195	115
POLYTRON PRODUCTS		
PVCS Corporate...*Source Code Control Syst.*	395	309
PVCS Personal	149	109
PVCS Network	Call	Call
PolyMake *Complete Make Utility*	149	109
PolyLibrarian *Library Manager*	99	73
PolyLibrarian II *Library Manaer*	149	109
PolyShell *UNIX-like Command Shell*	149	109
PolyXREF *Complete Cross Ref Utility*	219	169
PolyXREF *One language only*	129	99
PolyBoost *The Software Accelerator*	80	64
PolyDesk III *3rd Generation Desktop Org*	99	73
PolyDesk III *Add On Tools*	Call	Call
RYAN-McFARLAND PRODUCTS		
RM/FORTRAN...*2.4 New*	595	499
RM/FORTRAN...*UNIX or XENIX*	750	599
RM/COBOL...*DOS*	950	799
RM/COBOL...*UNIX or XENIX*	1250	999
RM/COBOL Run Time	250	199
RM/COBOL 85...*ANSI 85 COBOL*	1250	995
RM/COBOL 85 RUN TIME	300	249
RM/SCREEN...*for COBOL 85*	395	335
RM/NET +5...*COBOL & COBOL 85 Network*	300	249
SCREEN DESIGN		
Curses *by Lattice, UNIX screen designer*	125	99
with Source	250	199
Greenleaf Data Windows...*New*	225	169
with source	395	297
Panel Plus *by Round Hill, no royalties..New*	495	395
View Manager for C *by Blaise*	275	164
Vitamin C *by Creative Programming*	225	198
VC Screen *by Creative Programming*	100	81
Windows for C *Vermont Creative Software*	245	216
Windows for Data *includes Windows for C*	395	349
ZView *Data Management Consultants*	245	175
TEXT EDITORS		
Brief *from Solution Systems*	195	Call
Condor Editor *by Condor*	130	118
Edix *by Emerging Tech...Multi-screen*	195	159
Epsilon *by Lugaru Software, like EMACS*	195	149
FirsTime *by Spruce Technology, C syntax*	295	229
Kedit *by Mansfield, similar to Xedit*	125	99
LSE, the Lattice Screen Editor *Multi Wind.*	125	100
Vedit *by Compuview*	150	99
Vedit Plus *by Compuview*	185	129

3TJ8/87

PRICED TO SAVE YOU MONEY, SHIPPED FAST ANYWHERE.

dBx: dBASE AT THE SPEED OF C

Translates dBASE Applications to C

You dBASE™ programmers know what an expressive and readable language dBASE is. It's a very comfortable development environment. But the price is debased performance. Even compiled dBASE doesn't offer the speed that some users require from software these days. The kind of speed offered by software written in the C language.

The answer is dBx. It translates dBASE to C. It offers you a major competitive advantage over the next dBASE programmer. But not by converting everything you've ever written to C including you. Rather, keep right on writing in dBASE. Take every application all the way to completion. Then use dBx to translate it top to bottom to C source code and compile it.

There are other advantages. C is portable, even to other operating systems like UNIX™/XENIX™ as well as to the Macintosh or Amiga. dBx is the gateway for your applications to places where dBASE cannot go. (Library source needed for non MS-DOS).

dBx has its own file management utility for single user, but links to major C file managers—c-tree™ and dBC—for both compatibility with dBASE files and multi-user support. And you'll need a C compiler. We have everything to get you under way, including good advice. So call us when you're ready to put to C.

	List:	PC Brand:
dBx	**$ 350**	**$299**
with Library Source	**$ 550**	**$469**
with Full Source Code	**$1000**	**$855**

Call for special dBx Translation System pricing with includes dBx, C compiler, and File Manager.

ESSENTIAL C UTILITY LIBRARY

400 Functions, 30¢ Each

You've probably seen the speed and power of Essential's C function library without knowing it. Software greats have been using it for some time to give today's top products pizazz and panache.

Now grown to 400 functions Essential's collection features routines to produce pop-up menus, save and restore screens and windows to disk or memory in as little as 1/10th second, and the fastest video output available. Screen handling covers scrolling, paging, word processor functions to insert, delete and clear to end of lines or screen. Library has a complete set of 50 business graphics functions, 40 string handlers, 28 functions for printers, 18 for mice, 11 for time and date. Its DOS interfacing functions offer disk error trapping to keep DOS from intruding on your program, and directory and file management with file creation, file search using wild cards, plus acquisition and change of file attributes. Everything comes in source, including sample programs that demo library functions. We have versions with prebuilt libraries for all well-known C compilers, and a source code librarian is supplied for rolling your own.

	List:	PC Brand:
C Utility Library	**$185**	**$119**
Essential Graphics	**$250**	**$183**
Essential Communications	**$185**	**$125**
with Breakout Debugger	**$250**	**$189**

ZVIEW

Screen Design Aid

A complete package for screen design with full windows management as a bonus! Easy creation of screens with complex validation, such as range checking or required/optional data. Powerful Screen Paint utility for creating or editing applications screens. Built in security levels, set at run-time, control read or read/write access by field or screen. Automatic help screen processing for run-time aid per field or screen. Applications regain control during field tabbing, allowing run-time on-screen transaction processing or flow control. Run-time functions include Screen Read and Write with automatic transparent data conversion from screen image to data storage, Field Editing, Help Screen Processing, even a capability to change any field characteristic at run-time, plus Window Push Pop and Scroll. Versions for Lattice, Microsoft and Aztec C. Automatic free updates to registered users. No run-time royalties. List: **$245** PC Brand: **$175**

The CONDOR EDITOR

$99 WITH COMPILER PURCHASE

The developer of the Condor database system has come up with an unflappable editor with all the features you could hope for, especially a price that's light as a feather. No matter your language, the Condor Editor gives flight to your work with windows—tiled, overlapping, or split—to bring multiple files to the screen for simultaneous editing, full cut and paste between windows, and unlimited notepad buffers to store intermediate scraps. The editor has search capabilities to identify all files containing a sought for keyword and its location in those files, ideal for tracking down system-wide use of a function or variable. All DOS commands are available from within the editor. And it has either menued commands in desktop style or single keystroke commands once menus become obtrusive. Plus Condor claims the most extensive help facility of any editor should a new need come along. And for Condor 3 database management users, the Condor Editor cross-references command and form files. List: **$129.** PC Brand: **$118.** Together with compiler purchase the price drops to **$99.**

The DOLLAR is ON SALE!

Outside the United States? Our prices have plunged! Look what has happened to prices of programming tools in these major currencies:

		9/1/85	Press Time
Btrieve	DM	550	355
GSS Kernel	FF	3,200	2,260
Windows/Data	SF	600	389
C-Tree	Yen	78,500	46,980

We ship anywhere! *So take advantage of a bargain. Phone or telex your order now. For credit cards, we need card number, expiry date, cardholder's name and address. Or wire funds (see "Payment" below).* ***We ship immediately.***

BLAISE *C Tools Plus*

Nothing pays you back quicker than a function library, and Blaise has long been known as a great one. C Tools Plus, the top of the line, now has over 200 functions. Mature, tight, predominantly in C, they isolate hardware dependence, come in source and library, with no royalty. The rundown:

Screen Handling: either via BIOS or direct to video adapter; supports EGA, 43-line mode and multiple pages, twin monitors. Windows: any number, stackable, writable, wordwrapable, and erasable. Interrupts: well known for interrupt service routines in C, from which you can now access DOS functions. Directories: Create, change, or search directories, rename files, get and change their date/times. Memory: control DOS memory allocation, load a "child" process above your program, even create memory-resident programs—and remove them. Strings: international money, dates and character sets, plus superfunctions to perform several tasks with a single call. General: BIOS and DOS gates for direct access, memory copy, speaker control, it goes on. Our complete Blaise line:

	List:	PC Brand:
C Tools Plus	**$175**	**$117**
Asynch Manager	**$175**	**$117**
View Manager	**$275**	**$164**
Turbo Asynch Plus	**$100**	**$ 67**
Turbo Power Tools +	**$100**	**$ 67**

WINDOWS FOR DATA *M'soft Windows Compatible*

Windows for Data™ adds the pizazz and practicality of window partitioning for entering data in your application. Powerful field-level functions let you specify prompt string, field length, data type, screen location, picture, receiving variable, etc. Field options can require or prevent entry, cause beeping on invalid entry or overflow, attach field-specific help messages, functions to call for entry validation. You control which keys clear a field, jump to next or prior fields, quit, etc.

The Windows for C subset has all the windowing functions without data entry. Unlimited windows, defined in C structures for reference throughout your program, can be made either to pop up or permanently overwrite the screen. Routines scroll and highlight lists with arrow keys, read and scroll ASCII files both ways in windows. Logical treatment of video attributes permits unchanged programs to run on color or monochrome. Colors of windows are set individually. All functions are in separate modules. Only those used are linked. Only buffers holding on-screen or temporarily obscured windows occupy RAM; others released dynamically. Clear documentation.

Specify compiler:	List:	PC Brand:
Windows for Data	**$395**	**$349**
Windows for C	**$245**	**$216**

INTERACTIVE-C

Intrepreter, Editor, Debugger

Interactive-C™ is a fully integrated environment: a complete K&R interpreter bound to its own editor and "the best debugging facilities of any", says *Computer Language*, 2/87. Adjustable edit, command, and status windows, second screen for program output. Or twin CRT's. Can load object code of your compiled functions or commercial libraries. Immediate mode, syntax checking both when you type and run, and cursor that points precisely at errors. Debugger includes breakpoints, watchvalues, stepping options, interactive changing of variables. List: **$249**, PC Brand: **$219**.

BTRIEVE

ASK US ABOUT XTRIEVE & REPORT OPTION

Queen B-tree File Manager Abdicates Royalties

There's no longer a tithe to incorporate Btrieve™ in applications, a welcome proclamation if royalties would ruin your profit margins. Btrieve takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, forward and backward searching. It builds function call "commands" right into the language you use: interfaces to C, Pascal, BASIC, and COBOL, with sample programs in all four, come with each copy.

Btrieve has mainframe specifications! Its balanced-tree indexing scheme finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; indexes up to 255 characters; files of 4 billion bytes. Can even extend a file across two drives — even two hard disks!

Version 4.x speeds DOS interaction for large multiply-keyed files; enables variable length records of virtually any length; verifies accuracy (optionally) with read after write, useful in gritty environments; offers password and data encryption.

There's also Xtrieve, for Btrieve file inquiry and data manipulation, and Rtrieve for report writing. All three in versions for any network that supports the MS-DOS 3.1 file sharing function.

Ask for:	List:	PC Brand:
S0650	**$250**	**$195**
S0652 Network Version	**$595**	**$465**

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute its ownership. We will inquire for you about site licenses. Except as otherwise indicated or where "#" follows the Product Code, products may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM® PC, XT or AT under PC-DOS and require no more than 128k of RAM unless indicated. Non IBM machines using MS-DOS: contact manufacturer about precise differences so we can advise.

Returns: See box page one. Defective parts will be replaced. Please call for authorization to return a product for refund.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders or Literature, Call Us at....

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600

PC Brand, 150 5th Ave., New York, N.Y. 10011-4311

Telex: 667962 (SOFT COMM NYK)

Prices, terms, and specifications subject to change without notice.

4TJ8/87

CIRCLE NO. 233 ON READER SERVICE CARD

Developments for the systems professional

Amdek's 12.5-MHz System/286A

Mitsubishi Electronic's 16-MHz MP386

SYSTEMS

A series of 80386-based computers are being marketed by **Mitsubishi Electronics**. The **MP386 Series** operate at 16 MHz with zero wait states. Standard features include 32KB of cache memory (expandable to 64KB), sockets for an 80287 or 80387, 10 slots for expansion and custom configuration, and 5 half-height mass-storage devices. The MP386 will be sold through resellers who provide vertical-niche solutions and applications with emphasis on value-added features. With a 40MB disk-drive, $5,995.
Mitsubishi Electronics America, Inc., Computer Division, 991 Knox Street, Torrance, CA 90502; 213/515-3993
CIRCLE 303 ON READER SERVICE CARD

Based on a modular plug-in card architecture, a family of personal computers has been introduced by **Amdek Corporation**, a wholly owned subsidiary of Wyse Technology. The **System/88**, based on the 10-MHz Intel 8088, features a CPU of either 4.77 or 9.54 MHz. System/88 has a realtime clock, serial port, parallel port, 640KB of RAM, seven full-length expansion slots, and two half-height disk drive bays. With 360KB diskette drive, $1,115; and with a 20MB hard disk, $1,815.

The **System/286** is an 80286-based machine with 512KB RAM and all the features of the System/88 plus an additional half-height bay and a 190-watt power supply. With a 1.2MB diskette drive and drive controller, $1,795; and with a 20MB hard disk, $2,395.

The **System/286A**, based on the 12.5-MHz Intel 80286, features zero-wait-state memory architecture. The System/286A offers all the features of the System/286 plus an 8-MHz one-wait-state mode, LCD system status display, 1MB RAM (which is expandable to 16MB), an additional serial port, and a socket for a 6- or 10-MHz 80287. With a 1.2MB diskette drive, $2,450; and with a 40MB hard disk, $3,750.

The **System/386**, based on the 16-MHz Intel 80386, features six expansion slots and has all the features of the System/286 plus a dynamically switchable speed of 8 or 16 MHz, and sockets for an 80287 or 80387. With a single 1.2MB diskette drive, $3,850; and with a 40MB hard disk, $4,150.

The **System/386E**, is a highly expandable unit based on the 16-MHz Intel 80386 CPU, offering all the performance of the System/386 plus two additional half-height disk drive shelves and a 300-watt power supply. With a 1.2MB diskette drive, $4,250; and with a 40MB hard disk, $5,550.

Expandable 80386-based System/386E from Amdek

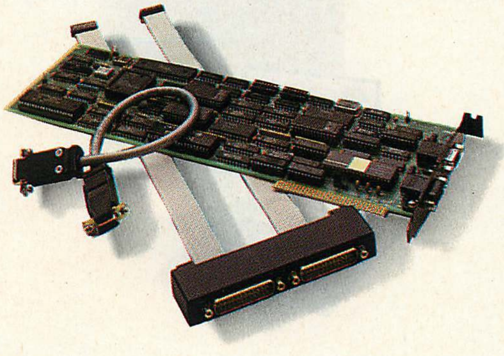
Amdek Corporation, 1901 Zanker Road, San Jose, CA 95112; 408/436-8570
CIRCLE 302 ON READER SERVICE CARD

A family of personal computers has been released by **NCR Corporation**. Based on NCR's Incremental Workstation Architecture, these computers feature flexible system configuration. This architecture isolates the processor on a single-slot card on the system bus, enabling users to upgrade to next-generation microprocessors. A Personality Card combines the functions of a video adapter, disk-drive controllers, extended memory, serial port, and parallel port on a single board. Discrete, functional modules can be added by the user; these modules contain expansion slots and storage devices that snap together, requiring no tools for installation. NCR offers a choice of keyboards.

The **PC916** is a 32-bit machine based on the 16-MHz Intel 80386. It features an advanced system design that isolates the microprocessor and the primary memory on separate, card-sized expansion boards that occupy two slots in the system's eight-slot bus. These boards are connected by a 32-bit extender bus, which allows direct 32-bit access between the processor and the memory for enhanced performance. The companion memory board comes with 2MB of 70-nanosecond dynamic RAM chips and can support as much as 268MB of physical memory and four gigabytes of virtual memory. The machine comes with a 5.25-inch 1.2MB diskette drive, Personality Card with EGA, and a clock/timer with battery. PC916 with 30MB hard disk, $6,353; with 44MB, $6,553; with 70MB, $7,553; or with 115MB, $8,653.

The **PC810** is NCR's AT-compatible based on the Intel 80286 running at 6 or 10 MHz. The PC810 uses the split-card architecture of the PC916, allowing users to upgrade to an 80386 in the future. The multifunction Personality Card is available in three varieties, providing EGA, color graphics/monochrome, or no graphics support. The PC810 comes configured with 640KB RAM (which can be expanded to 16MB) a 5.25-inch 1.2MB diskette drive, 720KB or 1.44MB 3.5-inch diskette drives, and a 20MB, 30MB, 44MB or 70MB hard-disk drives. Prices range from $2,950 for the basic model with no graphics adapter to $5,920 when configured with a 70MB hard disk.

PC810, a 10-MHz 80286-based system from NCR



AST-220 multitasking controller from AST Research, Inc.

The **PC710**, an entry-level 80286-based machine, also incorporates NCR's Incremental Workstation Architecture and runs at 6 or 10 MHz. The Personality Card supports a CGA or EGA. Prices range from $1,954 for the basic model with one 3.5-inch 720KB diskette drive and CGA to $2,870 for an added 20MB hard disk and EGA. Additional layers are priced as follows: with four expansion slots, $395; with a 3.5-inch 1.44MB diskette drive, $840; with 20MB hard disk, $1,310, with 3.5-inch 40MB tape backup, $1,390; with 5.25-inch 360KB diskette drive, $820; or with 5.25-inch 1.2MB diskette drive, $870.
NCR Corporation, Personal Computer Division, Dayton, OH 45479; 513/445-6240
CIRCLE 301 ON READER SERVICE CARD

CONNECTIONS

A multitasking controller that allows the PC, PC/XT, PC/AT and compatibles to operate in the Digital Equipment Corporation (DEC) VAX environment as windowing VT220 terminals is available from **AST Research, Inc.** A hardware-based solution, the **AST-220** provides a direct or remote connection between a PC and DEC's VAX or MicroVAX systems. The product consists of an add-on PC card, an RS-232 port mounting bracket for two VT220 ports, system software utilities, and an in-depth user's manual. The system software downloads onto the AST-220 card, which is driven by an Intel 80186 with 128KB of RAM. This leaves all but 16KB to 20KB of the PC's own memory free to run PC applications while using the AST-220. The system software provides a multitasking operating system with full windowing features. Five windows are available, all of which may operate simultaneously. The windows provide concurrent access to two separate DEC sessions, a DOS session, and two notepads. Kermit and XMODEM protocols are provided for use on PCs and VAX host systems. $595.
AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992; 714/863-1333
CIRCLE 306 ON READER SERVICE CARD

Digital Equipment Corporation (DEC) has announced the volume availability of the **IBM PC Network Integration Package**, which allows users of IBM PCs to participate in LANs using DEC's VAX/VMS Services for MS-DOS, a software package that allows VAX, MicroVAX, and VAXmate computers to act as application, data, and resource servers to groups of VAXmate and IBM PCs. These machines can then use the servers to share applications, data and resources, access information from remote systems on the network and apply it in industry-standard applications. A full package includes DEC's Ethernet controller module (DEPCA), dual-mode LK250 keyboard, DEC's mouse, Microsoft Windows, and VT220 terminal emulation. $1,195.

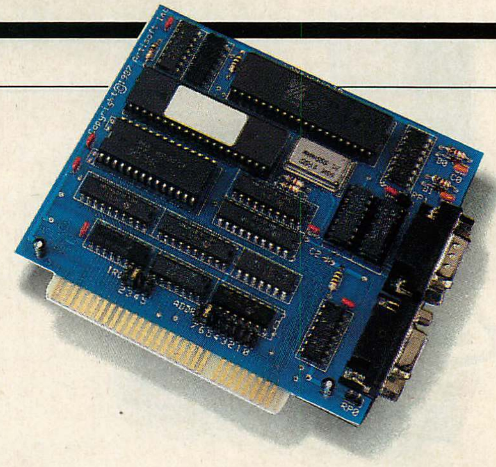
DEC's Network Integration Package

For customers who prefer the DOS command-line interface, DEC offers the **IBM PC Network Services Package**, which includes the same networking functionality, but does not support Windows or the VT220 Terminal Emulator and does not include the keyboard or mouse. $895.

DEC also has developed **DECnet/SNA Data Transfer Facility** (DTF), a layered software that allows users to move information and files between a DEC VAX-based system and a Systems Network Architecture (SNA) environment. With either VMS/SNA software or the DECnet/SNA Gateway as the link between the DEC and SNA environments, DTF can provide fast, efficient bidirectional file transfer capabilities using familiar commands and interfaces. Also announced is the **Advanced Program-to-Program Communications** (APPC) software to include the most current features available for LU6.2. This communications package enables programmers in an SNA environment to access information residing on a DEC network without the necessity of having a DEC user initiate the session. DTF server software ranges in price from $1,050 for the VAXstation 2000 to $21,000 for the VAX 8800. DTF server on IBM System/370 running the VMS operating system, $25,000. Client software from $450 for VAXstation 2000 to $9,000 for VAXstation 8800. APPC/LU6.2 ranges from $450 for the VAXstation 2000 to $9,000 for the VAX 8800.
Digital Equipment Corporation, Maynard, MA 01754-2571; 800/344-4825
CIRCLE 305 ON READER SERVICE CARD

A wholly owned subsidiary of Novell, Inc., **CXI, Inc.**, has released four connectivity products. **PCOX 3270** allows IBM PC and PS/2 models on a LAN to access as many as 40 concurrent mainframe sessions. The **PCOX/COAX-MUX**, a full-length, high-performance coaxial interface board, connects a PC or PS/2 and its LAN to the 3299 interface of an IBM 3274 or 3174 cluster controller. PCOX/COAX-MUX works in conjunction with CXI's **PCOX/GW-3270 server**, to provide the gateway to a mainframe for PCs and PS/2s on a LAN. All PCOX 3270 coaxial and remote

VINES/386 network operating system from Banyan Systems



LANtastic 2.0 from Artisoft, Inc.

LAN-to-mainframe gateway software will now support the NetWare SPX LAN interface. Depending on which PCOX gateway is used, from 5 to 64 host sessions can be distributed to PCs and PS/2s on a Novell LAN. CXI's five LAN workstation packages provide micro-to-mainframe functionality to PCs and PS/2s on LANs equipped with PCOX coaxial and remote gateways. **PCOX/TWO** is a 3270 micro-to-mainframe software package that runs on PCs or PS/2s equipped with a PCOX, IBM, IRMA, or AST coaxial board. It provides the capability to access one host printer session concurrently with eight one-host sessions or one PC session. PCOX/COAX-MUX, $1,100; PCOX/GW-3270 server, $850; PCOX/TWO, $400; PCOX/TWO, LAN version, $200.
CXI, Inc., 1157 San Antonio Road, Mountain View, CA 94043-1069; 800/225-7269; in California, 415/969-1999
CIRCLE 309 ON READER SERVICE CARD

A version of a network operating system designed to run on the Compaq Deskpro 386 has been introduced by **Banyan Systems, Inc.** The **VINES/386** package is designed to take full advantage of the 80386's 32-bit, protected-mode operation and full addressing capabilities. VINES/386 provides a high-performance migration path from Banyan's VINES/286. The software maintains full compatibility with Banyan's high-performance Motorola 68000-based network servers: the Banyan/BNS and Banyan/DTS. VINES/386 software will include protocol support for communicating with minicomputers and mainframes or with larger networks that include additional Banyan servers. For connection to IBM mainframes, a VINES/386-server will allow PCs attached to the network to emulate 3270 terminals over SNA/SDLC protocols. File transfer capabilities will be available to all PCs attached to a server through a variety of integrated and third-party software products. VINES/386 will provide communications with asynchronous hosts via VT100, VT52, IBM3101, and TTY emulation. File transfer through Kermit protocol also will be available. $3,995.
Banyan Systems Inc., 115 Flanders Road, Westboro, MA 01581; 617/898-1000
CIRCLE 307 ON READER SERVICE CARD

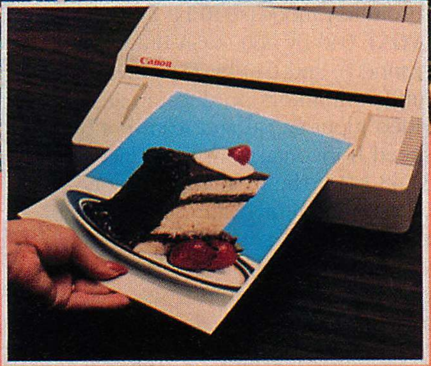
A second generation asynchronous gateway has been announced by **Novell, Inc.** The **NetWare Asynchronous Connection Service** (NACS) provides LANs running Novell's NetWare operating system with improved access to asynchronous resources; an optional hardware add-on product allows off-the-shelf terminal emulation software to run with the new gateway. Some NACS features are transparent resources connection, ability to connect to any number of asynchronous gateways by resource name and ability to handle the details of finding an available modem or minicomputer and then setting of the connection, and support for up to nine asynchronous sessions at one time. NACS supports Novell's X.25 gateway. NACS, $1,095; NetWare Asynchronous Board, $149.

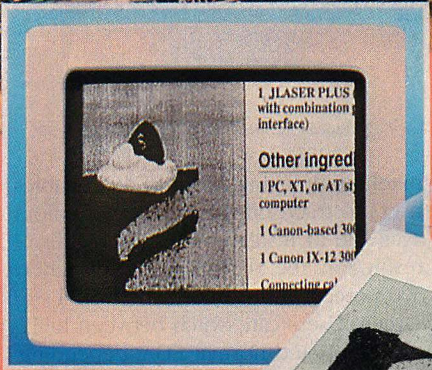
Novell's NetWare Asynchronous Connection Service

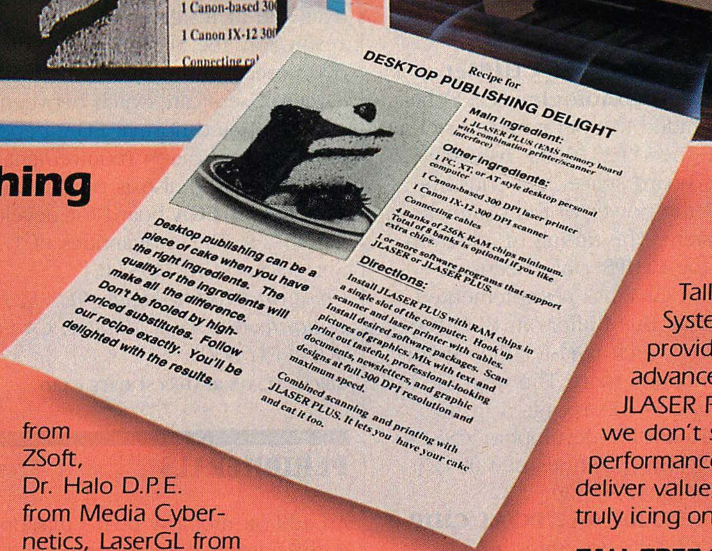
Novell's **NetWare Asynchronous Remote Bridge** will connect multiple remote NetWare LANs to a local NetWare LAN over standard telephone lines. This bridge will allow remote PCs to gain access to a local NetWare LAN. The bridge functions transparently, allowing a user to transfer data to a remote LAN using a command similar to one used to send data locally. Single-line, $395; multiline, $995.
Novell, Inc., 122 E. 1700 South, Provo, UT 84601; 800/453-1267; in Utah, 801/379-5900
CIRCLE 308 ON READER SERVICE CARD

A networking product with an on-board network processor is available from **Artisoft, Inc.** With an enhanced speed as fast as 1.4 Megabits per second (Mbps), **LANtastic 2.0** is compatible with NETBIOS and be used with the IBM PC Network, Novell's Advanced Netware or any other NETBIOS-compatible LAN. Because the on-board processor implements the entire NETBIOS, LANtastic manages all network traffic, thus freeing up the PC to perform computing tasks. LANtastic is a bus network and can communicate directly with any other PC without any intervening hardware such as hubs, splitters, expanders, or tap boxes. When two computers want to communicate on the bus at the same time, LANtastic uses a proprietary CSMA/CD method that can detect a network packet collision within two microseconds. LANtastic starter kit, including two half-slot network cards, cable, bus terminators and NETBIOS software, $399; individual cards and cable, $199.
Artisoft, Inc., 3550 N. First Avenue, Suite 330, Tucson, AZ 85719; 602/293-6363
CIRCLE 310 ON READER SERVICE CARD

An automatic dialing synchronous/asynchronous modem has been introduced by **Network Software Associates, Inc.** (NSA). The board-level **AdaptModem 201/212** modem incorporates a built-in synchronous data-link control (SDLC) adapter. In synchronous







It makes desktop publishing a piece of cake!

Tall Tree Systems introduces another breakthrough in desktop publishing with JLASER PLUS. We've combined a 2 MB EMS memory board and an interface to both a Canon®-based laser printer and scanner. JLASER PLUS increases the performance of both devices and gives you a low-cost solution to the limitations you've been experiencing with them.

Furthermore, the same memory that is made available to your printer and scanner is also available for all your other conventional applications. You get system memory, expanded LIM memory, extended memory in an AT-type machine, RAM Disk and print spooler — all in a single slot!

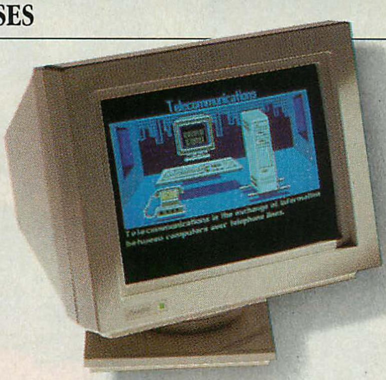
Supporting JLASER PLUS is a host of software packages, such as PC Paintbrush + from ZSoft, Dr. Halo D.P.E. from Media Cybernetics, LaserGL from Software Express, Ventura Publisher from Xerox, Page Builder from White Sciences, Le Print from Le Baugh Software, Fancy Font and Fancy Word from SoftCraft, Inc., and many more to be announced.

It takes a technological innovator like Tall Tree Systems to provide a major advancement like JLASER PLUS. And we don't stop at performance. We also deliver value, which is truly icing on the cake.

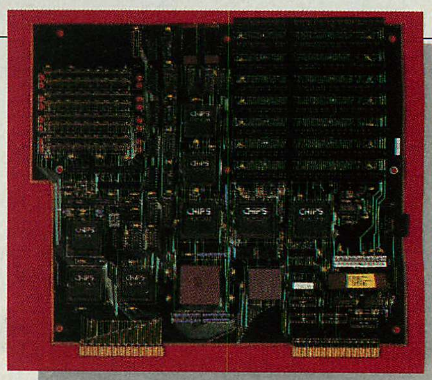
TALL TREE SYSTEMS
2585 E. Bayshore Road
Palo Alto, CA 94303
(415) 493-1980
Telex: 9102404041

CIRCLE NO. 194 ON READER SERVICE CARD

TALL TREE SYSTEMS



Monitor/732, Amdek's VGA-compatible color monitor



386/AT Board with CS8230-20 AT/386 CHIPSet from Chips and Technologies

mode, it functions as a Bell 201C modem at 2400 bps (bits per second), and in asynchronous mode, it functions as a Bell 212A modem at 300, 1200, or 2400 bps, and supports the Hayes AT command set. NSA offers a full line of communications software programs for synchronous communications, including, PC-to-PC, PC-to-minicomputer, and PC-to-mainframe. The AdaptModem's automatic call control (ACC) software module is a standard feature that provides a number of dialing facilities for synchronous applications including automatic dialing, answering, and redialing; automatic phone-line and modem testing; modem configuration options; and a call directory. $995.
Network Software Associates, Inc., 22982 Mill Creek, Laguna Hills, CA 92653; 714/768-4013

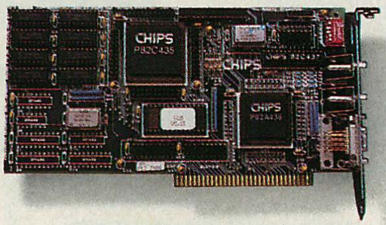
CIRCLE 311 ON READER SERVICE CARD

TECHNOLOGY

The **CS8230-20 AT/386 CHIPSet Chips and Technologies** (C&T) has been upgraded to operate at more than twice the speed (20 MHz), use one-third less board space, and consume one-third the power of the original PC/AT system. The timing of all seven chips in the CHIPSet has been improved and C&T has reimplemented the two Data Bus Buffers in Bi-CMOS technology, which is faster than its bipolar ALS counterpart. The upgraded CHIPSet also is incorporated in the **386/AT Board** for developers. CS8230-20, $141.50 (in quantities of 1,000); 386/AT Board, $2,995.00

The **CS8220-10/12 PC/AT CHIPSet** from C&T incorporates an advanced memory controller that provides the dynamic bus-clock-switching function, enabling a system to run at full speed for on-board memory, and then switch to half-speed for all off-board memory and I/O. It also supports very large memory configurations from 1MB to 4MB on the system board. This CHIPSet also is incorporated in the **286/AT Board** for developers. CS8220-10/12, $51.60 (in quantities of 1,000); 286/AT Board, $1,495.00.

The **82C437 SharpScan** from C&T is an IBM EGA-compatible graphics chip that provides 1,128-by-560 line resolution for clearer text. Using a



C&T's SharpScan EGA with 82C437 graphics chip

pixel-multiplexing function that enables users to trade off the number of colors available (4 instead of 16) for higher resolution, the SharpScan can display 300 percent more information on a screen. Users can switch between high resolution and normal EGA, depending on their application requirements. The SharpScan chip also is present on the **SharpScan EGA** board for developers. 82C437, $6.70 (in quantities of 1,000); SharpScan EGA, $695.00.
Chips and Technologies, Inc., 521 Cottonwood Drive, Milpitas, CA 95035; 408/434-0600

CIRCLE 312 ON READER SERVICE CARD

PERIPHERALS

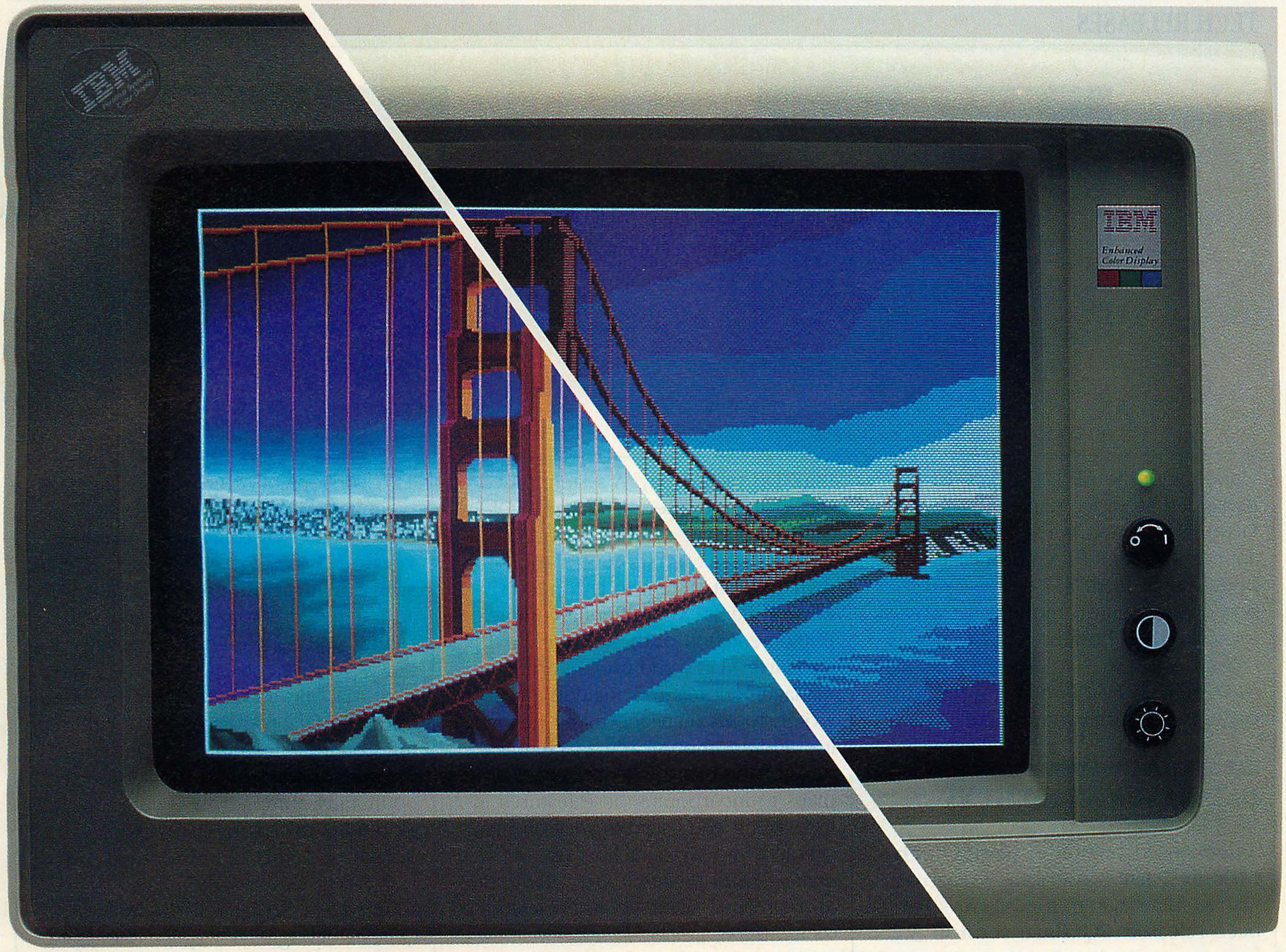
An add-on memory board that operates at the advanced RAM storage and speed levels of the Compaq Deskpro 386 is available from **Computer Peripherals, Inc.** The **386 Memoire** operates in the host's 32-bit mode at speeds as fast as 16 MHz with zero wait states. A 25-pin connector matches the unique built-in connector in the Deskpro 386, providing access to its 32-bit memory. 1MB, $845; 2MB, $1,395.
Computer Peripherals, Inc., 2635 Lavery Court, Suite 5, Newbury Park, CA 91320; 800/854-7600, in California, 805/499-5751

CIRCLE 315 ON READER SERVICE CARD

A family of color and monochrome computer monitors fully compatible with the IBM Video Graphics Array (VGA) standard has been introduced by **Amdek Corporation**. Amdek's **Monitor/432** monochrome monitor and **Monitor/732** color monitor offer three resolution modes: 640 by 480 pixels (VGA), 640 by 400 (CGA double-scanned), and 640 by 350 (EGA). The Amdek monitors produce an analog video input signal and feature a high-speed horizontal scan frequency of 31.48 KHz. The Monitor/432 monochrome monitor features a 14-inch diagonal display, antiglare flat screen with a choice of paper white or amber phosphor. The Monitor/732 color monitor has a 12-inch antiglare screen with 0.28mm-dot pitch. Both monitors feature vertical-scan frequency of 50, 60, and 70 Hz, and a tilt-and-swivel stand. Monitor/432, $245; Monitor/732, $625.
Amdek Corporation, 1901 Zanker Road, San Jose, CA 95112; 408/436-8570

CIRCLE 317 ON READER SERVICE CARD

An enhanced version of the **AT_Meg** memory expansion board has been announced by **PBJ, Inc.** The AT_Meg provides up to 8MB of extended or expanded memory on a single board. The board runs at zero wait states on 6/8-MHz machines and at one wait state on 10- or 12-MHz machines. Using single in-line memory modules (SIMMs), the AT_Meg provides both high-density packaging through surface mount technology, plus the ability to upgrade using PC board plug-in modules. In-



SIGMAVGA BRIDGES THE GRAPHICS GENERATION GAP.

True VGA Graphics . . . Delivered.
Forget the promises or the confusion. Sigma Designs cuts through the smokescreen of competing claims with the definitive VGA solution, *including Mode 13.* Introducing SigmaVGA, the high resolution graphics display adaptor that provides 100% VGA BIOS-level compatibility for the PC, XT, AT and compatibles. SigmaVGA delivers the future generation of PS/2 graphics software without compromising your ability to run existing programs. And it's available right now!

More Graphics Flexibility Than Ever Before.
There's no need to sacrifice your existing hardware investment or access to popular graphics modes in exchange for VGA capability. With SigmaVGA, you can have it all:

More Colors—A demonstrated spectrum of 256 colors out of a palette of 256,000. *Now it's easier than ever before to represent complex information, execute designs or create intensely vivid presentations.*

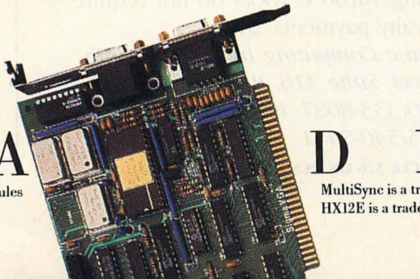
More Displays and Emulations—SigmaVGA drives digital, analog and multisync monitors and provides EGA, CGA, MDA and Hercules Graphics emulations.

200,000+ EGA Boards Installed.
A proven leader in the field, Sigma Designs has been supplying advanced graphics hardware to major OEM customers. Now you can take advantage of this superior technology.

To find out more about SigmaVGA or other advanced Sigma Designs products, see your local dealer or call 415-770-0100.

SIGMA VGA
VGA Enhanced Display Modes · 320 x 200 x 256 Colors (Mode 13) Out of a palette of 256,000 colors · 640 x 480 x 16 Colors More resolution than ever before · 720 x 400 x 16 Colors or Gray Scales Easy reading 9 x 16 matrix in text mode
Emulation Modes EGA, CGA, MDA, Hercules
Monitors Supported · IBM PS/2 Monochrome Display 8503 · IBM PS/2 Color Display 8512, 8513 · NEC MultiSync (analog and digital modes) · Princeton HX12E · Compatible analog and digital monitors, both monochrome and color

Sigma Designs Inc.
46501 Landing Parkway
Fremont, CA 94538
Telex 171240
Fax 415-770-0100

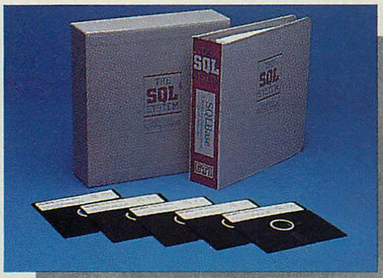


S I G M A D E S I G N S

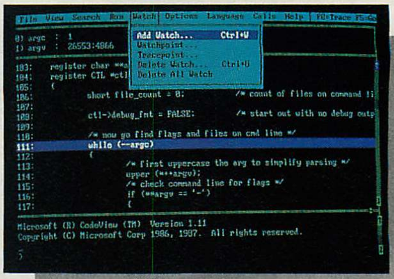
IBM is a registered trademark of Internatinal Business Machines. Hercules Graphics is a trademark of Hercules Computer Technology, Inc.

MultiSync is a trademark of NEC Corporation.
HX12E is a trademark of Princeton Graphic Systems, Inc.

CIRCLE NO. 202 ON READER SERVICE CARD



SQLBase 3.2 from Gupta Technologies, Inc.



CodeView debugging screen in Microsoft C Optimizing Compiler 5.0

cluded with the AT_Meg board is an Expanded Memory Manager (EMM) Driver that supports expanded memory applications and a RAM-disk program. Prices range from $579 for a 2MB board using 256KB modules to $1,999 for a full 8MB board.
PBJ Inc., 503 E. 40th Street, Paterson, NJ 07504; 201/523-8663
CIRCLE 314 ON READER SERVICE CARD

DATABASE MANAGEMENT

An enhanced version of **Gupta Technologies, Inc.**'s distributed database management system for PC LANs has been released. **SQLBase version 3.2** features a complete implementation of the structured query language (SQL) and runs on IBM PCs under DOS 3.2. It transforms one or more IBM PC/ATs into high-performance, relational database servers that can be accessed by multiple PCs running on the IBM Token-Ring Network. The system also operates on any NETBIOS-compatible networks, including Novell's Advanced Netware and 3Com's 3Plus. It includes a C application programming interface for Microsoft Windows. Multiuser development toolkit, $1,995 per server; single-user version, $995.
Gupta Technologies, Inc., 1040 Marsh Road, Suite 240, Menlo Park, CA 94025; 415/321-9500
CIRCLE 318 ON READER SERVICE CARD

A software technology licensing agreement has been reached between **Applied Data Research, Inc.** (ADR) and **Software Systems Technology, Inc.** (SST). Under this agreement, ADR will acquire the software technology for **XDB**, SST's relational database management system for IBM PCs, LANs and other minicomputer and microcomputer systems. With this agreement, ADR obtains a worldwide license to develop software products incorporating XDB technology. A system based on the structured query language (SQL), XDB gives ADR a full-function SQL capability for its **DATACOM/DB** database management system in LAN and PC environments. XDB offers a complete implementation of the current ANSI standard. XDB's end-user information management features include a report writer, forms manager, menu generator, and an import/export facility for transferring data from other files. ADR plans to incorporate these features in its existing end-user information management products, such as DATAQUERY.
Applied Data Research, Inc., Route 206 and Orchard Road, CN-8, Princeton, NJ 08540-0008; 201/874-9000
CIRCLE 319 ON READER SERVICE CARD
Software Systems Technology, Inc., 7100 Baltimore Avenue, Suite 206, College Park, MD 20740; 301/779-5486
CIRCLE 320 ON READER SERVICE CARD

SOFTWARE DEVELOPMENT

A library for building programs with Borland's Turbo C compiler is available from **Blaise Computing, Inc.** The **Turbo C TOOLS** library of compiled C functions allows the programmer to have full control over the PC, the video environment, and the file system. The library comes with well-documented source code that can be studied, emulated, or adapted to specific needs. Features include the use of function prototyping, prebuilt libraries for all memory models, cleanly organized header files, and a comprehensive, fully indexed manual. Runtime files generated using Turbo C TOOLS do not require royalty payments. $129.
Blaise Computing Inc., 2560 Ninth Street, Suite 316, Berkeley, CA 94710; 800/333-8087; in California, 415/540-5441
CIRCLE 326 ON READER SERVICE CARD

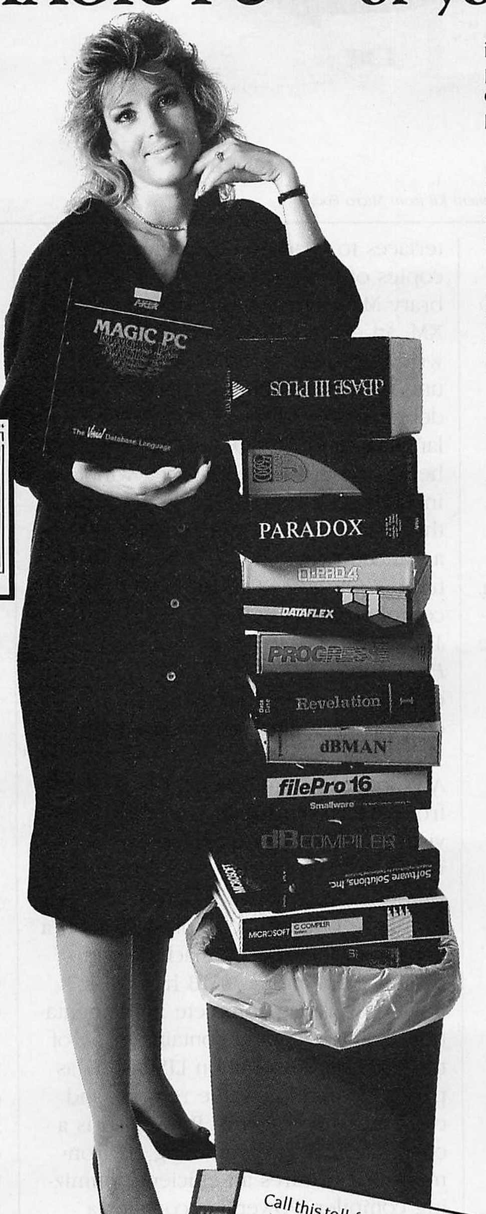
Two versions of **Microsoft C** language compiler, **Microsoft QuickC** and **Microsoft C Optimizing Compiler version 5.0**, are available from **Microsoft Corporation**. QuickC comes with an integrated editor, compiler, MAKE facility, and source-level debugger, extensive documentation, which includes a tutorial and a context-sensitive, online help facility. Microsoft C Optimizing Compiler includes the QuickC in-memory compiler and the advanced optimizing compiler to enable programmers to develop prototypes quickly and then optimize them. Version 5.0 gives users the choice of two source-level debuggers: the integrated QuickC debugger for quick debugging while prototyping, and the enhanced CodeView debugger (version 1.11) for more complex debugging. QuickC and Version 5.0 are completely source- and object-level compatible; QuickC emits CodeView-compatible executable files. Microsoft QuickC, $99; Microsoft C Optimizing Compiler 5.0, $450; update from 4.0 to 5.0, $75.

Microsoft also announced the first release of its **MS-OS/2 Software Development Kit** (SDK), which allows software developers to begin the progression of moving applications software to the new MS-OS/2 environment for 80286 and 80386-based personal computers. The developer's toolkit contains a prerelease version of the MS-OS/2 system kernel and technical specifications for the kernel and the MS-OS/2 LAN Manager. Also included are MS-OS/2 versions of Microsoft's macro assembler (MASM) and C language compilers, the Microsoft CodeView debugger, and other software development tools (including a program editor). In addition to a software and technical seminar, the development toolkit includes a one-year license for Microsoft's Direct Information Access Line (DIAL) system, an electronic mail technical support service, and a one-year

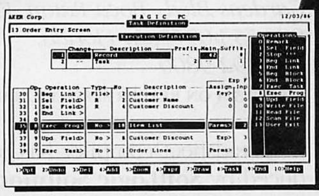
Make $500/hr.

Now develop DBMS applications 10 times faster for only $199 with MAGIC PC — or your money back!

Database programmers, why waste your time hacking out code?
Imagine how much faster and more profitable you'd be if you could whip up powerful database applications without the time-consuming coding pains . . .
Introducing Magic PC from Aker, your professional dream come true. It's not *another line-by-line* syntax treadmill like any DBMS or 4GL.
Finally you can program as quickly as you design, while you delegate all the mundane and redundant coding tasks to Magic PC.

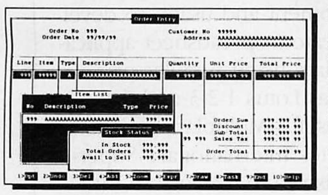


Program 10 times faster



Develop relational database applications 10 times faster using a visual design-driven interface. Instead of writing mountains of "how to" procedural code, you quickly place your program design specs in Execution Tables and Magic PC's engine executes them automatically. Don't lose any more time editing and debugging programs by hand.

Incredible Zoom power



Magic PC's phenomenal Zoom power magically co-executes related programs through nested Zoom windows smoothly with auto data scrolling in all directions. While Zooming, query and transfer data across windows or even Zoom deeper.

No more maintenance!

Change your programs on the fly without any manual maintenance responsibility. Magic PC automatically updates your changes online since all the data describing your design (data dictionary, programs and menus) make up a single file, self-maintaining Integrated Library.

Magic PC does it all

Design your entire database application with only one comprehensive development system. Generate both online programs (screens, windows, menus), as well as batch programs (reports, updates, import/export, etc.) with full color and graphics. You no longer fall between the cracks dealing with separate and inconsistent programming utilities.

Free LAN features

Develop multi-user applications for local area networks with Magic PC's automatic support for file and record locking security.

Quick prototyping

Prototype a complete working application in just hours and get immediate customer feedback to finalize the design. It's a true time-saver.

Stand-alone runtime

Distribute your applications and protect your design with a low cost runtime engine. It has the friendliest end-user visual interface you've ever seen with built-in, menu-driven and syntax-free data retrieval power.

Jeff Duntemann, PC Tech Journal:

"Magic PC is probably the best integrated database application generator that we have seen . . . very smooth system, and smoothness comes at a premium these days." Also recommended by PC Magazine, PC World, PC Week, Computer Language, Data Based Advisor and many more around the world.

Try it for $19^95

If you develop database applications for a living, you can't afford not to try Magic PC for yourself right now. For $19.95 you'll get the Magic PC Tutorial software and documentation for hands-on evaluation, complete with a step-by-step guide to develop an Order Entry sample application in just a few hours.

Magic PC ~~$695~~ $199

No kidding! For a limited time only, save almost $500 off the $695 list price, and get the complete unprotected Magic PC software for only $199 at our special introductory non-resale price.

Money back guarantee

Even at $199 you can't go wrong with our no-risk guarantee: keep it only if it makes magic for you, or we'll buy it back within 30 days less $19.95 restocking fee.

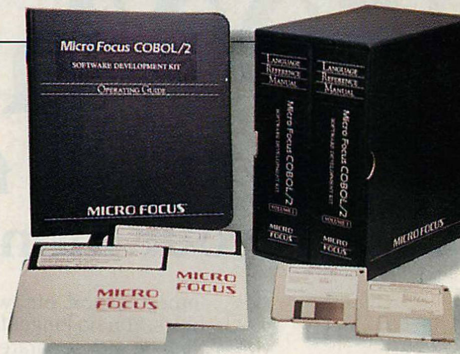
System Requirements: IBM PC, XT, AT, PS/2 and 100% compatible, PC-DOS 2.0 or later, 512K, hard disk. All trademarks acknowledged.

$199 ~~$695~~

Call this toll-free number now with your credit card or COD charge. Dealer pricing available.
800-345-MAGIC • in CA 714-250-1718
Or send this order coupon with your payment to:
Aker 18007 Skypark Cir. B2, Irvine, CA 92714
___ Magic PC at ~~$695~~ $199 (add $10 P&H)
___ Magic PC Tutorial at $19.95 (add $5 P&H)
In CA add applicable tax
Name ______
Company ______
Address (no POB) ______
City/St/Zip ______
Phone ______
Check enclosed, or charge: VISA MasterCard AMERICAN EXPRESS
Acct. No. 1 2 3 4 5 6 7 8 9 0
Exp. Date ___/___
Name on card ______
Signature ______

MAGIC PC
The *Visual* Database Language
by AKER

CIRCLE NO. 153 ON READER SERVICE CARD



COBOL/2 Software Development Kit from Micro Focus

PowerLisp AI environment from MicroProducts, Inc.

subscription to the *Microsoft Systems Journal*. MS-OS/2 SDK, $3,000.
Microsoft Corporation, 16011 N.E. 36th Way, P.O. Box 97017, Redmond, WA 98073-9717; 800/426-9400; in Washington and Alaska, 206/882-8088

CIRCLE 321 ON READER SERVICE CARD

A pre-release version of the **Lattice OS/2 C Compiler** has been announced by **Lattice, Inc.** This compiler allows mixed memory models and direct calls to the operating system. It eliminates the 64KB static data limitation, and it gives an option to make the default integer size 32 bits. The compiler features an extensively reworked library and expanded manuals, which include information designed to help programmers take advantage of OS/2's advanced features. Available only as an upgrade to Lattice C 3.2, this version includes all prerelease updates and a copy of the production version, $150.
Lattice Inc., 2500 S. Highland Avenue, Lombard, IL 60148; 800/533-3577; in Illinois, 312/916-1600

CIRCLE 327 ON READER SERVICE CARD

Designed for DOS and OS/2 applications, the **COBOL/2 Software Development Kit** (SDK) has been released by **Micro Focus**. The Micro Focus COBOL/2 compiler, featured in SDK, features the ability to bring existing COBOL code from a non-DOS environment, support of nine COBOL dialects, and a broad implementation of the COBOL language. COBOL/2 compiles ANSI 74 and ANSI 85 COBOL, OS/VS and VS COBOL II. Micro Focus Level II COBOL, RM/COBOL, DG Interactive COBOL, IBM PC COBOL version 1, and IBM's Systems Application Architecture standard. The COBOL/2 compiler also eases the tasks of migrating COBOL source code through its 32-bit addressing architecture. Standard features include network support, Micro Focus ANIMATOR visual debugging tool, call interfaces to C and other languages, and copies of the Microsoft Linker and Library Manager. The SDK also includes XM, an extended memory program for writing protect mode applications under DOS; Xilerator, a object code debugger that displays both assembly language and COBOL; Panels, which helps in creating overlapping windows in COBOL; the COBOL/2 Editor; and the Resident Run-Time System, with advanced program management features. COBOL/2 SDK, $1,500; COBOL/2 compiler alone, $900.
Micro Focus, 2465 E. Bayshore Road, Palo Alto, CA 94303; 800/872-6265; in California, 415/856-4161

CIRCLE 322 ON READER SERVICE CARD

A PC-based AI environment is available from **MicroProducts, Inc.** that offers virtual memory. **PowerLisp** can develop and run professional 60MB applications with the speed, power, and performance of a symbolic processor on a PC/AT or 80386-based machine with 3MB of RAM and a 40MB hard disk. PowerLisp is the complete implementation of InterLisp and contains many of the features of Common LISP, such as packages, multiple-value returns, and catch-and-throw forms. PowerLisp is a complete LISP programming environment that features an efficient optimizing compiler. PowerLisp contains a debugging program, a structure editor, and Masterscope, a static-program analyzer that allows the developer to find and edit every place in the program where a given function is called or where references are made to specific variables, object, or properties. Program editing features allow for editing and individually recompiling functions, including the ability to put breakpoints in functions as well as trace them, and print variable and stack bindings.

Also available is **Power-Ex**, a expert system shell developed at Stanford University under the name EMYCIN, is an optional development tool. 286 version of PowerLisp, $1,195; with a 3MB expansion board, $1,695; 80386 version, $1,695; Power-Ex (when purchased with PowerLisp), $500.
MicroProducts, Inc., 370 W. Camino Gardens Blvd., Boca Raton, FL 33432; 800/553-0777; in Florida, 305/392-9800

CIRCLE 324 ON READER SERVICE CARD

A spreadsheet compiler has been introduced by **SoftLogic Solutions, Inc.** The program, **@Liberty**, enables users to create and distribute executable spreadsheet applications that can be run without the original spreadsheet program for as little as $10 per user. Using proprietary software technology, @Liberty separates the tasks of spreadsheet development and usage so developers can create a spreadsheet application or template using popular programs (such as Lotus 1-2-3 and Symphony, Microsoft's Multiplan, and Computer Associates International's SuperCalc4) and distribute executable, runtime files. End-users can run the @Liberty-compiled spreadsheet without the original spreadsheet program to enter and modify specified data, calculate results and export data back to the original spreadsheet. All formulas are invisible to and are protected from modification or destruction by the user. Licensing plan in which the developer can distribute @Liberty-compiled spreadsheets and documentation for 10 users, $99.95; refill pack and 15 more user licenses, $99.95.
SoftLogic Solutions, Inc., One Perimeter Road, Manchester, NH 03103; 800/272-9900; in New Hampshire, 603/627-9900

CIRCLE 325 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

HARD FACTS:

We're Programmer's Connection, your best one-stop source for quality programmer's development tools for IBM personal computers and compatibles. Here are some important facts you should know about us and other dealers in our industry.

FACT:
FREE Shipping. Shipping to U.S. customers is FREE via UPS Ground. If you want your order shipped via an express service, we'll only charge you the shipping carrier's standard rate with no special fees. Some dealers charge extra for shipping and then add rush charges for shipping via express services. Others may advertise "free" shipping but make up for it by charging extra handling fees.

FACT:
Credit Cards. We'll charge your credit card only when we actually ship your order. Some dealers would charge your credit card at the time you place your order. This could leave you waiting for your shipment for weeks or months while they use your money interest-free.

FACT:
Discounts. We discount all software products — even special order items. Every product in our advertised price list is shown with its list price and discounted price. We want you to know exactly how much you'll save on every product. We don't try to fool you by discounting some products and charging full retail for others.

FACT:
Consistent Prices. We extend the same current prices to every customer regardless of where they see our ad. Some dealers vary prices in different ads and then ask you to mention which one you saw. This technique allows them to charge you the highest prices possible.

FACT:
No Hidden Charges. The discount prices you see on the next two pages are all you pay. We don't charge extra for UPS Ground shipping, credit cards, COD orders, purchase orders, sales tax (except Ohio) or special handling (except for non-Canadian international orders).

FACT:
Guarantees. We offer FREE 30-day no-risk return guarantees and 30-day evaluation periods on most of our products. Some dealers have no return options while others often charge restocking fees of 15% or more.

FACT:
Quality Products. Our product line consists of hundreds of high quality software development tools specifically for IBM Personal Computers and compatibles. While some dealers try to carry every software product ever written, we carry only those that meet our very high standards for quality and value.

FACT:
Latest Versions. The products we carry are the latest versions and come with the same manufacturer's technical support as if buying direct. While some dealers may participate in the software gray market, we're authorized to sell every product we carry.

FACT:
Large Inventory. We have one of the largest inventories of programmer's development products in the industry. Most orders are shipped within 24 hours. And if we don't have a product in stock, we'll get it for you fast.

FACT:
Meticulous Packaging. We'll give your shipment the extra protection needed to reach you in the best possible condition. First we'll protect your products from moisture by wrapping them in plastic. Then we'll insulate your box with high quality bubble-wrapping instead of the messy styrofoam chips that many other dealers use. Finally, we'll double-tape your box for extra strength.

FACT:
Independence. Since we're not directly affiliated with any software publisher or developer, we can give you sound, unbiased advice. Unlike some dealers who have a special interest in promoting only certain products, we'll give you an objective look at the products we carry.

FACT:
Noncommissioned Staff. Our courteous sales staff is always ready to help you. And if you aren't sure about your needs, our knowledgeable technical staff can give you sound, objective advice. Because they are noncommissioned, you won't be pressured into making a purchase.

As you can see, we're different from the other dealers in our industry. Our customers keep coming back because we consistently provide the highest quality service and the lowest prices. So call us today and experience the differences for yourself.

PROGRAMMER'S
QUALITY INTEGRITY
Service
SUPPORT PRICE
CONNECTION

Turn the page for our product list and ordering information.

ai - expert systems

	List	Ours
1st-CLASS *by Programs in Motion*	495	399
EXSYS *Development Software by EXSYS*	395	309
EXSYS Runtime System	600	469
Logic-Line Series *All varieties by Thunderstone*	CALL	CALL

ai - lisp language

GCLISP *Golden Common LISP by Gold Hill*	495	CALL
GCLISP 286 Developer *by Gold Hill*	1190	CALL
Microsoft LISP *Common LISP*	250	149
QNIAL *Combines LISP & APL by NIAL Systems*	CALL	CALL
TransLISP PLUS *from Solution Systems*	195	125

ai - prolog language

Arity Combination Package	1095	979
Expert System Development Pkg	295	229
File Interchange Toolkit	50	44
PROLOG Compiler & Interpreter	650	569
Screen Design Toolkit	50	44
SQL Development Package	295	229
Arity PROLOG Interpreter	295	229
Arity Standard Prolog	95	77
LPA microPROLOG *All Varieties*	CALL	CALL
MPROLOG Language Primer *LOGICWARE*	50	45
MPROLOG P500 *by LOGICWARE*	495	395
MPROLOG P550 *by LOGICWARE*	220	175
Turbo PROLOG *by Borland Intl*	100	64
Turbo PROLOG Toolbox *by Borland Intl*	100	64

ai - smalltalk language

Smalltalk/V	99	84
EGA/VGA Color Option	50	45
Goodies Diskette	50	45
Smalltalk/Comm	50	45

ai - texas instruments

Arborist *Decision Tree Software* *New*	595	519
PC Scheme *Lisp*	95	84
Personal Consultant Easy	495	435
Personal Consultant Image *New*	495	435
Personal Consultant Online *New*	995	869
Personal Consultant Plus	2950	2589
Personal Consultant Runtime	95	84

ada language

AdaVantage *by Meridian Software Systems* *New*	795	735
AdaVantage Utility Packages *New*	CALL	CALL
DOS Environment Package *New*	50	47
Janus/ADA C Pak *by R&R Software*	95	84
Janus/ADA D Pak *by R&R Software*	1250	1059
Janus/ADA ED Pak *by R&R Software*	395	349

apl language

APL*PLUS/PC *by STSC*	595	424
APL*PLUS/PC Spreadsheet Mgr *by STSC*	195	139
APL*PLUS/PC Tools Vol 1 *by STSC*	295	199
APL*PLUS/PC Tools Vol 2 *by STSC*	85	58
ATLAS*GRAPHICS *by STSC*	450	329
Financial/Statistical Library *by STSC*	275	189
Pocket APL *by STSC*	95	69
STATGRAPHICS *by STSC*	795	579

assembly language

386 ASM/LINK *Cross Asm by Phar Lap*	495	389
8088 Assembler *w/Z-80 Translator by 2500 AD*	100	89
ASMLIB *Function Library by BC Assoc*	149	125
asmTREE *B-Tree Dev System by BC Assoc*	395	329
Cross Assemblers *Various by 2500 AD*	CALL	CALL
EZASM *by C Source* *New*	70	64
Microsoft Macro Assembler *New Version*	150	93
Norton Utilities *by Peter Norton*	100	CALL
Norton Utilities (Advanced)	150	99
Turbo Debugger *by Speedware* *New*	89	79
Turbo Editasm *by Speedware*	99	84
Visible Computer: 80286 *New*	100	89
Visible Computer: 8088 *by Software Masters*	80	64

basic language

87 Software Pak *by Hauppauge*	180	149
db/Lib *for QuickBASIC by AJS Publishing* *New*	99	89
EXIM Services Toolkit *by EXIM*	50	45
Finally *by Komputerwerks*	99	85
MACH 2 *by Micro Help*	69	55
Microsoft QuickBASIC *$20 Rebate Offer*	99	63
QBase *Relational Database by Crescent* *New*	89	79
Quick-Tools *by BC Associates*	130	109
QuickPak *by Crescent Software*	69	59
Scientific Subroutine Library *by Peerless*	125	99
Screen Sculptor *by Software Bottling*	125	91
Stay-Res *by MicroHelp*	69	55
True Basic *w/Run-time* *Special Price*	200	99
True Basic	100	79
Run-time Module	100	79
Various Utilities	50	41
Turbo BASIC Compiler *by Borland Intl*	100	64

blaise products

ASYNCH MANAGER *Specify C or Pascal*	175	135
C TOOLS PLUS	175	135
LIGHT TOOLS *for Datalight C* *Special Price*	100	65
PASCAL TOOLS	125	99
PASCAL TOOLS 2	100	79
PASCAL TOOLS & TOOLS 2	175	135
RUNOFF *Text Formatter*	50	45
TURBO ASYNCH PLUS	100	79
TURBO C TOOLS	129	CALL
TURBO POWER TOOLS PLUS	100	79
VIEW MANAGER *Specify C or Pascal*	275	199

borland products

EUREKA *Equation Solver*	167	105
Reflex & Reflex Workshop	200	128
Reflex *Data Base System*	150	89
Reflex Workshop	70	45
Sidekick & Traveling Sidekick	125	85
Sidekick	85	57
Traveling Sidekick	70	45
Superkey	100	64
Turbo BASIC Compiler	100	64
Turbo BASIC Database Toolbox *New*	100	64
Turbo BASIC Editor Toolbox *New*	100	64
Turbo BASIC Telecom Toolbox *New*	100	64
Turbo C Compiler *(Call for support products)*	100	64
Turbo Database Toolbox	70	41
Turbo Editor Toolbox	70	41
Turbo Gameworks Toolbox	70	41
Turbo Graphix Toolbox	70	41
Turbo Jumbo Pack	300	219
Turbo Lightning	100	64
Turbo PASCAL Numerical Methods Toolbox	100	64
Turbo PASCAL and Tutor	125	85
Turbo PASCAL	100	64
Turbo Tutor	40	24
Turbo PROLOG Compiler	100	64
Turbo PROLOG Toolbox	100	64
Word Wizard	70	47
Word Wizard and Turbo Lightning	150	94

c compilers

C86PLUS *by Computer Innovations*	497	375
DeSmet C *w/Debugger & Large case*	209	184
DeSmet C *w/Debugger only*	159	138
Eco-C *Complete System by Ecosoft*	140	119
Lattice C Compiler *vers. 3.2 from Lattice*	500	265
Mark Williams Let's C w/csd	125	99
Microsoft C Compiler w/CodeView *New version*	450	269
Microsoft QuickC Compiler *New*	99	63
Optimum-C *by Datalight*	139	105
Turbo C Compiler *by Borland*	100	64
Uniware 68000/10/20 Cross Compiler *by SDS*	995	829

c interpreters

C-terp *by Gimpel, Specify compiler* *New Version*	300	219
C Trainer *with Book by Catalytix*	122	87
Instant C *by Rational Systems* *New Version*	500	369
Introducing C *by Computer Innovations*	125	99
Run/C *by Age of Reason*	120	69
Run/C Professional *by Age of Reason*	250	145

c utilities

C++ *by Guidelines w/version 1.1 kernel*	195	172
Csharp Realtime Toolkit *by Systems Guild* *New*	600	539
c-tree & r-tree Combo *by FairCom*	650	519
c-tree *ISAM File Manager*	395	315
r-tree *Report Generator*	295	239
Data Windows *by Magus Inc* *New*	245	209
with Source Code *New*	595	499
dBx *dBASE to C Translator by Desktop AI*	350	299
with Source Code *New*	550	469
Flash-up Windows *by Software Bottling*	90	78
GraphiC *Color version by Sci Endeavors*	350	282
GRAFLIB *by Sutrasoft*	175	159
HALO *Graphics by Media Cybernetics*	300	205
HALO Development Pkg *for Microsoft*	595	389
The HAMMER *by OES Systems*	195	129
PANEL *Forms Management by Roundhill*	295	215
PANEL/TC *for Turbo C by Roundhill* *New*	129	95
PANEL Plus *by Roundhill*	495	395
PC Lint *by Gimpel Software*	139	99
PLOTHP *by Sutrasoft*	175	159
RTC PLUS *Fortran to C by Cobalt Blue*	450	399
Scientific Subroutine Library *by Peerless*	175	135
TE Text Editor *source by Sub Systems* *New*	95	85
Vitamin C *by Creative Programming*	225	158
VC Screen *Forms Designer*	100	79
Zview *by Data Mgmt Consultants*	245	139

cobol language

COBOLspII *by Flexus*	395	329
FPLIB *for Realia COBOL by BC Associates*	149	129
Micro Focus COBOL *See Micro Focus Section*		
Microsoft COBOL *See Microsoft Section*		
PCDT *by Pro-Code* *New*	995	895
Realia COBOL with RealMENU *New Version*	1145	899
Realia COBOL *New Version*	995	783
RealCICS	995	783
RM/COBOL *by Ryan-McFarland*	950	CALL
RM/COBOL 85 *by Ryan-McFarland*	1250	CALL
SCREENIO *by Norcom* *New*	400	379
screenplay *for COBOL by Flexus*	175	129

css products

Combo Package *by Custom Software Systems* *New*	199	175
PC/SPELL *Spelling Checker*	49	45
PC/TOOLS *UNIX-like Utilities*	49	45
PC/VI *vi Editor*	149	99

debuggers & profilers

386 DEBUG *Cross Debugger by Phar Lap*	195	129
Advanced Trace-86 *by Morgan Computing*	175	115
Codesmith-86 *by Visual Age*	145	98
DSD87 *by Soft Advances*	125	79
MiniProbe *by Atron*	395	369
Periscope I *with Board by Periscope*	345	289
Periscope II *with NMI Breakout Switch*	175	139
Periscope II-X *Software only*	145	105
Periscope III *8 MHz version*	995	799
Periscope III *10 MHz version*	1095	899
The PROFILER *with Source Code by DWB*	125	89
TURBOsmith *Source debugger for Turbo Pascal*	69	59
The WATCHER *Profiler by Stony Brook*	60	51

disk utilities

Back-It *by Gazelle Systems*	100	89
Disk Optimizer *by Softlogic Systems*	60	55
FASTBACK *by 5th Generation Systems*	179	129
Vcache *by Golden Bow Systems* *New*	50	47
Vopt *by Golden Bow Systems* *New*	50	47
Vfeature *by Golden Bow Systems* *New*	80	74
Vfeature Deluxe *by Golden Bow Systems* *New*	120	111
XenoCopy-PC *by XenoSoft*	80	69

dos utilities

Command Plus *by ESP Software*	80	69
FANSI-CONSOLE *by Hersey Micro*	75	62
MicroHelp Utilities *by MicroHelp*	59	49
Norton Commander *by Peter Norton*	75	CALL
OPAL *Shell Language by Software Factory*	99	89
Q-DOS II *by Gazelle Systems*	70	59
Taskview *by Sunny Hill Software*	80	55

essential products

C Utility Library	185	119
Essential Comm Library *with Debugger*	250	189
Essential Comm Library *Software Only*	185	125
Breakout Debugger Only *Any language*	125	89
Essential Graphics	250	183

forth language

CFORTH *Native Code Compiler by LMI*	300	229
Forth/83 Metacompiler *Specify Target*	750	599
PC/Forth *by Laboratory Microsystems*	150	109
PC/Forth+ *by Laboratory Microsystems*	250	199
Advanced Color Graphics Support	100	74
Enhanced Graphics Support	200	148
Intel 8087 Support	100	74
Interactive Symbolic Debugger	100	74
Native Code Optimizer	100	74
Software Floating Point	100	74
UR/Forth *by LMI*	350	279
Object Module Libraries	500	395

fortran language

50 MORE: FORTRAN *by Peerless Scientific*	125	95
ACS Time Series *by Alpha Computer Service*	495	389
AUTOMATED PROGRAMMER *by KGK Automated* *New*	CALL	CALL
Essential Graphics *by Essential Software*	250	183
For-Winds *by Alpha Computer Service*	90	69
Forlib-Plus *by Alpha Computer Service*	70	44
FORTLIB *by Sutrasoft*	125	109
FORTRAN Addendum *by Impulse Engr*	95	85
FORTRAN Addenda *by Impulse Engr*	165	138
GRAFLIB *by Sutrasoft*	175	159
HALO *Graphics by Media Cybernetics*	300	205
I/O PRO w/No Limit Library *by MEF*	250	219
Microcompatibles Combo Package	240	215
Grafmatic	135	117
Plotmatic	135	117
Microsoft FORTRAN *w/CodeView*	450	269
No Limit Library *by MEF Environmental*	129	109
Numerical Analyst *by MAGUS*	295	249
PANEL *by Roundhill Computer Systems*	295	215
PLOTHP *by Sutrasoft*	175	159
RM/FORTRAN *by Ryan-McFarland*	595	CALL
RTC PLUS *Fortran to C by Cobalt Blue*	450	399
Scientific Subroutine Lib *by Peerless*	175	135
Statistician *by Alpha Computer Service*	295	235
Statlib.GL: *by Peerless* *New Version*	295	239
Statlib.TSF: *by Peerless* *New Version*	295	239
Strings & Things *by Alpha Computer Service*	70	45

greenleaf products

Greenleaf Comm Library	185	125
Greenleaf Data Windows Library	225	155
with Source Code	395	249
Greenleaf Functions	185	125

help utilities

HELP/Control *by MDS*	125	99
On-line Help *from Opt-Tech*	149	99
SoftScreen/HELP *by Dialectic Systems*	195	149

lattice products

Lattice C Compiler *ver 3.2 from Lattice*	500	265
with Library Source Code	900	495
C Cross Reference Generator	50	37
with Source Code	200	139
C-Food Smorgasbord *Function Library*	150	95
with Source Code	300	179
C-Sprite *Source Level Debugger*	175	119
Curses *Screen Manager*	125	85
with Source Code	250	169
dBC *Specify dBC II or dBC III*	250	169
with Source Code	500	356
dBC III Plus	750	594
with Source Code	1500	1184
LMK *Make Facility*	195	138
RPG II Combo *All three items below*	1100	875
RPG II Compiler *No Royalties*	750	625
RPG II SEU *Screen Entry Utility*	250	199
Sort/Merge	250	199
RPG II Screen Design Aid Utility	350	309
SecretDisk *File Encryption Utility*	120	88
SideTalk *Resident Communications*	120	88
SSP/PC *Scientific Subroutine Library*	350	269
Text Management Utilities	120	88
TopView Toolbasket *Function Library*	250	178
with Source Code	500	356

metagraphics products

LightWINDOW/C *for Datalight C*	95	79
FontWINDOW	95	79
FontWINDOW/PLUS	275	229
MetaWINDOW *No Royalties*	195	159
MetaWINDOW/PLUS	275	229
TurboWINDOW/C *for Turbo C*	95	79
TurboWINDOW/Pascal *for Turbo Pascal*	95	79

micro focus products

Micro Focus Level II COBOL *w/Animator*	495	395
Level II COBOL	349	279
Level II Animator	195	155
Micro Focus Level II COBOL/ET for UNIX	CALL	CALL
Micro Focus Personal COBOL	149	119
Micro Focus Professional COBOL	2000	1595
Micro Focus VS COBOL/XENIX	1495	1195

Micro Focus Support Products:

COBOL/IQ *Ad hoc Report Writer*	495	395
COBOL/IQ *for DOS 3.X Networks*	995	795
FORMS-2	295	235
SOURCEWRITER	995	795

microport products

System V/386 Combination *New*	799	699
386 Runtime System *New*	199	169
386 Software Development System *New*	499	429
Text Preparation System *New*	199	169
386 Unlimited License Kit *New*	249	209
DOSMerge386 *Run DOS and UNIX together* *New*	CALL	CALL
System V/AT Combination	549	465
AT Runtime System	199	169
AT Software Development System	249	209
Text Preparation System	199	169
AT Unlimited License Kit	249	209
DOSMerge286 *Run DOS and UNIX together* *New*	149	129

microsoft products

Microsoft BASIC Compiler *for XENIX*	695	419
Microsoft BASIC Interpreter *for XENIX*	350	209
Microsoft C Compiler *w/CodeView* *New Version*	450	269
Microsoft COBOL Compiler *with COBOL Tools*	700	429
for XENIX	995	609
Microsoft FORTRAN *Optimizing Compiler/CodeView*	450	269
Microsoft FORTRAN *for XENIX*	695	419
Microsoft Learning DOS	50	36
Microsoft LISP *Common LISP*	250	149
Microsoft MACH 10 *with Mouse & Windows*	549	369
Microsoft MACH 10 *Board only*	399	279
Microsoft Macro Assembler *New Version*	150	93
Microsoft Mouse *for IBM PS/2* *New*	175	114
Microsoft Mouse *Bus Version*	175	114
Microsoft Mouse *Serial Version*	195	124
Microsoft muMath *Includes muSIMP*	300	179
Microsoft Pascal Compiler	300	179
for XENIX	695	419
Microsoft QuickBASIC *$20 Rebate Offer*	99	63
Microsoft QuickC *New*	99	63
Microsoft Sort	195	125
Microsoft Windows	99	63
Microsoft Windows Development Kit	500	299
Microsoft Word	450	269

modula-2 language

EXE2LNK *MASM Interface by PMI* *New*	49	45
Macro2 *Macro preprocessor by PMI* *New*	89	79
ModBase *by PMI* *New*	89	79
MODULA-2 Apprentice Pkg by LOGITECH	99	79
MODULA-2 Magic Pkg by LOGITECH	99	79
MODULA-2 ROM Pkg & Cross RT Debugger	299	239
MODULA-2 Window Pkg by LOGITECH	49	39
MODULA-2 Wizard's Pkg by LOGITECH	199	159
Repertoire *by PMI*	89	75

mouse products

LOGIMOUSE BUS *with PLUS Pkg by LOGITECH*	119	98
with PLUS & PC Paintbrush	149	119
with PLUS & CAD Software	189	153
with PLUS & CAD & Paint	219	179
LOGIMOUSE C7 *with PLUS Pkg, Specify Connector*	119	98
with PLUS & PC Paintbrush	149	119
with PLUS & CAD Software	189	153
with PLUS & CAD & Paint	219	179
Microsoft Mouse *See Microsoft Section*		

other languages

ACTOR *by Whitewater Group* *New*	495	419
CCS MUMPS *Single-User by MGlobal*	60	50
CCS MUMPS *Single-User Multi-Tasking*	150	129
CCS MUMPS *Multi-User*	450	359
Marshal Pascal *by Marshal Language Systems*	189	165
Pascal-2 *by Oregon Software*	395	325
Personal REXX *by Mansfield Software*	125	99
SNOBOL4+ *by Catspaw*	95	80

other products

Carbon Copy *by Meridian Technology* *New*	195	179
Dan Bricklin's Demo Pgm *by Software Garden*	75	57
Dan Bricklin's Demo Tutorial	50	45
Fast Forward *by Mark Williams* *New*	70	59
First Publisher *with Mouse from Logitech* *New*	CALL	CALL
Informix *All Varieties by Informix*	CALL	CALL
Instant Replay *by Nostradamus*	150	CALL
Mace Utilities *Paul Mace Software* *New*	99	89
MKS Toolkit *w/vi Editor by MKS* *New Version*	139	114
MicroTEX *Typesetting from Addison Wesley*	295	CALL
Net-Tools *by BC Associates*	149	129
OPT-Tech Sort *by Opt-Tech Data Proc*	149	99
PC/TOOLS *by Custom Software*	49	45
Screen Machine *by MicroHelp*	79	59

phoenix products

Pasm86 *Macro Assembler version 2.0*	195	108
Pdisk *Hard Disk & Backup Utility*	145	99
Pfantasy Pac *Phoenix Combo*	995	595
Pfinish *Execution Profiler*	395	209
Pfix86plus *Symbolic Debugger*	395	209
PforCe *Specify C Compiler*	395	209
PforCe++ *Specify C Compiler and C++*	395	209
Plink86plus *Overlay Linker*	495	275
Pmaker *Make Utility*	125	78
Pmate *Macro Text Editor*	195	108
Pre-C *Lint Utility*	295	154
Ptel *Binary File Transfer Program*	195	108

polytron products

PolyBoost *The Software Accelerator*	80	64
PolyDesk III	99	72
PolyDesk III Archivist	50	42
PolyDesk III Cryptographer	50	42
PolyDesk III Talk	70	52
PolyLibrarian *Library Manager*	99	89
PolyLibrarian II *Library Manager*	149	129
PolyMake *UNIX-like Make Facility*	149	129
PolyShell	149	129
Polytron C Beautifier	50	45
PolyXREF *Complete Cross Ref Utility*	219	185
PolyXREF *One language only*	129	109
PVCS Corporate *Version Control System*	395	329
PVCS Personal	149	129

program mgmt utilities

Interactive EASYFLOW *by Haventree*	150	125
PrintQ *by Software Directions*	89	84
Quilt Computing Combo Package	250	199
QMake *Program Rebuild Utility*	99	79
SRMS *Software Revision Mgmt System*	185	159
Source Print *by Aldebaran Labs*	97	75
TLIB *Version Control System by Burton*	100	89
Tree Diagrammer *by Aldebaran Labs*	77	67

raima products

dbQUERY *Single-User Query Utility*	195	129
Single-User with Source Code	495	389
Multi-User	495	389
Multi-User with Source Code	990	699
dbVISTA *Single-User DBMS*	195	129
Single-User with Source Code	495	389
Multi-User	495	389
Multi-User with Source Code	990	699

sco products

Complete XENIX System V *by SCO*	1295	994
Development System	595	499
Operating System *Specify XT or AT*	595	499
Text Processing Package	195	144
Lyrix *by SCO*	595	449
SCO Professional *1-2-3 Workalike for XENIX*	795	595
SCO XENIX-NET	595	495

softcraft products

Btrieve *ISAM Mgr with No Royalties*	245	184
Xtrieve *Query Utility*	245	184
Report Option *for Xtrieve*	145	99
Btrieve/N *for Networks*	595	454
Xtrieve/N	595	454
Report Option/N *for Xtrieve/N*	345	269

text editors

Brief & dBrief Combo *from Solution Systems*	275	CALL
Brief	195	CALL
dBrief *Customizes Brief for dBASE III*	95	CALL
Epsilon *Emacs-like editor by Lugaru*	195	147
KEDIT *by Mansfield Software*	125	98
Micro/SPF *by Phaser Systems*	175	139
Microsoft Word	450	269
PC/VI *by Custom Software Systems*	149	105
SPF/PC *by Command Technology Corp*	CALL	CALL
Vedit Plus *by CompuView*	185	128

turbo pascal utilities

ALICE *Interpreter by Software Channels*	95	66
DOS/BIOS & Mouse Tools *by Quinn-Curtis*	75	67
Flash-up Windows *by Software Bottling*	90	78
MACH 2 *for Turbo Pascal by Micro Help*	69	55
MetraByte D/A Tools *by Quinn-Curtis*	100	89
Science & Engrg Tools *by Quinn-Curtis*	75	67
Screen Sculptor *by Software Bottling*	125	91
Speed Screen *by Software Bottling*	35	32
System Builder *by Royal American*	150	129
IMPEX *Query Utility*	100	89
Report Builder	130	115
TDebugPLUS *by TurboPower Software*	60	49
Tmark *by Tangent Systems* *New*	80	69
Turbo EXTENDER *by TurboPower Software*	85	64
Turbo OPTIMIZER *by TurboPower*	75	65
Turbo OPTIMIZER *with Source Code*	125	108
Turbo Professional *by Sunny Hill*	70	45
Turbo.ASM *by BC Associates* *New*	100	89
TurboHALO *from IMSI*	129	98
TurboPower Utilities *by TurboPower*	95	78
TurboRef *by Gracon Services*	50	45
TURBOsmith *Source Debugger by Visual Age*	69	59
Universal Graphics Library *by Quinn-Curtis*	150	119

wendin products

Operating System Toolbox	99	79
PCNX *Operating system*	99	79
PCVMS *Similar to VAX/VMS*	99	79
Wendin-DOS *Multitasking DOS* *New*	99	85
XTC *Text Editor w/Pascal source*	99	75

xenix/unix products

Btrieve *ISAM File Mgr by SoftCraft*	595	454
C-terp *by Gimpel, Specify compiler*	498	379
c-tree *ISAM Mgr by FairCom*	395	329
dBx *with Library Source by Desktop AI*	550	469
Desqview *from Quarterdeck* *New*	100	85
DOSIX *Console Version by Data Basics*	399	349
DOSIX *User Version by Data Basics*	199	179
Informix *All Varieties by Informix*	CALL	CALL
Micro Focus Products *See Micro Focus Section*		
Microport Products *See Microport Section*		
Microsoft Products *See Microsoft Section*		
PANEL Plus *by Roundhill Computer Systems*	795	535
REAL-TOOLS *Binary Version by PCT*	149	89
Library Source Version	399	289
Complete Source Version	999	729
RM/COBOL *by Ryan-McFarland*	1250	CALL
RM/FORTRAN *by Ryan-McFarland*	750	CALL
SCO Products *See SCO Section*		

Call or write for our FREE comprehensive price guide.

LOWEST PRICES

Due to printing lead times, some of our current prices may differ from those shown here. Call for latest pricing.

FREE SHIPPING

Orders within the USA (including Alaska & Hawaii) are shipped FREE via UPS. Express shipping is available at the shipping carrier's standard rate with no rush fees or handling charges. To avoid delays when ordering by mail, please call first to determine the exact cost of express shipping.

CREDIT CARDS

VISA and MasterCard are accepted at no extra cost. Your card is charged when your order is shipped. Mail orders please include credit card expiration date and authorized signature.

CODs AND POs

CODs and Purchase Orders are accepted at no extra cost. No personal checks are accepted on COD orders. POs with net 30-day terms (with initial minimum order of $100) are available to qualified US accounts only.

SALES TAX

Orders outside of Ohio are not charged state sales tax. Ohio customers please add 6% Ohio tax or provide proof of tax-exemption.

INTERNATIONAL ORDERS

Shipping charges for International and Canadian orders are based on the shipping carrier's standard rate. Since rates vary between carriers, please call or write for the exact cost. International orders (except Canada), please include an additional $10 for export preparation. All payments must be made with US funds drawn on a US bank. Please include your telephone number when ordering
Due to government regulations, we cannot ship to all countries.

VOLUME ORDERS

Volume orders may qualify for additional discounts. Call us for special pricing.

SOUND ADVICE

Our knowledgeable technical staff can answer technical questions, assist in comparing products and send you detailed product information tailored to your needs.

30-DAY GUARANTEE

Most of our products (excluding books) come with a 30-day documentation evaluation period or a 30-day return guarantee. Please note that some manufacturers restrict us from offering guarantees on their products. Call for more information.

MAIL ORDERS

Please include your telephone number on all mail orders. Be sure to specify computer, operating system and any applicable compiler or hardware interface(s). Send mail orders to:

Programmer's Connection
Order Processing Department
136 Sunnyside Street
Hartville, OH 44632

USA	**800-336-1166**
CANADA	**800-225-1166**
OHIO & ALASKA (Collect)	**216-877-3781**
TELEX	**9102406879**
EASYLINK	**62806530**
INTERNATIONAL	**216-877-3781**
CUSTOMER SERVICE	**216-877-1110**

Hours: Weekdays 8:30 AM to 8:00 PM EST.

PROGRAMMER'S Service CONNECTION
QUALITY INTEGRITY SUPPORT PRICE


The more th
the more they
1971. Kansas tornado blows out DEC™ PDP-IIs. Files saved on 3M data cartridge tape.*
1972. Moonlighting programmer deprograms company's production records. Files saved on 3M data cartridge tape.
1973. Fastidious janitor turns off IBM® 370. Files saved on 3M data cartridge tape.
1978. Colorado electric storm jolts Wangs.® Files saved on 3M data cartridge tape.
1979. Little Stevie Fong flips floppies out father's office window. Files saved on 3M data cartridge tape.
1980. Temporary help permanently dumps accounting records on Apple III. Files saved on 3M data cartridge tape.
1984. Hard disk fails in soft market; brokers panic. Files saved on 3M data cartridge tape.
1985. Sal's Diner. Dropped eggs scramble Macs.™ Files saved on 3M data cartridge tape.
1987. Delivery boy delivers IBM PS/2™ swiftly and abruptly to the sidewalk. Files saved on 3M data cartridge tape.
3M
data cartridge technology

ings change,
stay the same.

1975. Head of the office trips, pulls plug on HP® 3000. Files saved on 3M data cartridge tape.

1976. Unnamed computer does the unmentionable. Files saved on 3M data cartridge tape.

1977. Office fire bakes Apple® IIs. Files saved on 3M data cartridge tape.

1981. Circuit breaker flips, floppies flop. Files saved on 3M data cartridge tape.

1982. Head crash proves fatal to Lisa.™ Files saved on 3M data cartridge tape.

1983. Thief sneaks away with Osbornes.® Files saved on 3M data cartridge tape.

1990. Somewhere, somehow, on some computer, data will crash. Files will be saved on 3M data cartridge tape.

Computers come, and computers go.

But when they go out abruptly, you've always been able to rely on 3M data cartridge tape for backup.

In fact, 80% of the people who back up on data cartridge tape back up on 3M.

Why?

We've been covering computer and human errors almost longer than computers and humans have erred together.

Not only did we invent and patent data cartridges, we've continually patented the innovations that make data cartridge backup the best way to back up data.

Which is something that's never changed.

Call (800) 423-3280 for a brochure.

3M

Worldwide Sponsor 1988 Olympic Games

Trademarks/owner: DEC/Digital Equipment Corporation; IBM, Personal System/2/International Business Machines Corp.; HP/Hewlett-Packard; Apple, Lisa/Apple Computer Inc.; Wang/Wang Laboratories; Osborne/Novell, Inc.; Mac/MacIntosh Laboratory, Inc.

*True to life stories. CIRCLE NO. 164 ON READER SERVICE CARD ©3M 1987

MAN BUYS INBOARD 386; ENDS UP ON DESERT ISLAND.

When John Middleton's 1-2-3 applications got to be more than his IBM PC AT® could handle, he had two choices.

Spend $6000 for a new 386 system.

Or spend $2000 for an Intel® Inboard 386 system.

So he got the best of both worlds. With the Intel Inboard 386, he got the full power of a 386 system. Without compromising one bit on performance or compatibility. Then, with the leftover $4000, he spent two glorious weeks on the island of Bora Bora.

How can we give you so much for so much less? Simple. If you have an IBM PC AT or compatible, you already own ⅔ of a 386 system. And when you install an Inboard 386, you get the rest of it.

Besides price, how do they compare? According to a recent *Infoworld* product review, "Inboard has the best computing speed of all micro systems we have tested to date," including all other 386 systems.

The reason Inboard is so fast is because of its zero wait state cache and 32 bit memory.

And there's also a special socket for adding the 80387-16 math coprocessor giving you even greater performance.

Software compatibility is unsurpassed as well.

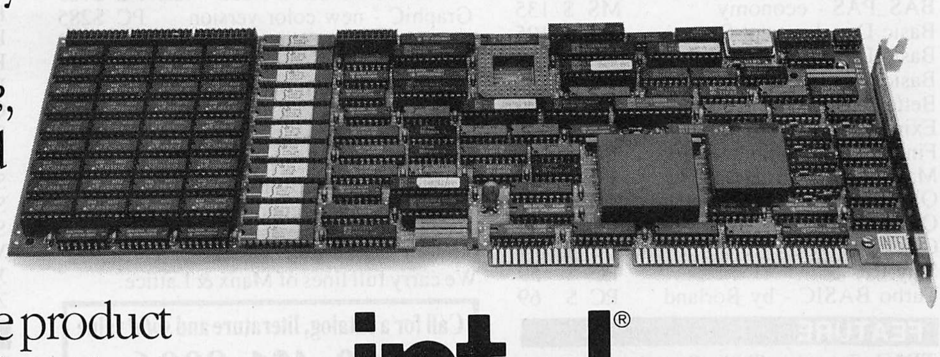
According to *PC Week*, "The Inboard 386 proved perfectly compatible with a standard IBM PC AT and every software product we tested." It's compatible with advanced software, too, including the new 386 control software for multitasking. And lots of developers are already using Inboard to create OS/2® applications.

But maybe the real question is can a $2000 system be more reliable than a $6000 one? The answer is yes. Because it's built by the same company that designed the 80386 microprocessor and it's backed by a five-year warranty.

If you still need more information on which system is best, call us at (800) 538-3373. Or contact your local dealer.

Then we suggest you find a nice quiet place, clear your mind of all distractions, and think it over.

Maybe a nice quiet place like Bora Bora.



intel®

CIRCLE NO. 187 ON READER SERVICE CARD

TECH NOTEBOOK 80

GERRY KAPLAN

Underline Fix for the EGA

The underlining capability of the EGA can be restored in color mode with this terminate-and-stay-resident program.

Text-based applications are most often used on the IBM monochrome display. Users of word processors appreciate its sharp characters formed by a 9-by-14-pixel matrix, as well as its ability to display underlined and bolded text. An Enhanced Color Display connected to an EGA can provide virtually the same level of sharpness with an 8-by-14-pixel character cell, and it can do this in 16 colors. But on a color monitor, the EGA does not display underlined text.

The EGA is capable of underlining, and it does so when connected to a monochrome monitor. The hardware is designed to produce underlining for characters with video attributes of 1 or 9 for the foreground and zero for the background, provided that the contents of the Underline Location Register contains a value between 0 and 13. This value is the scan line on which the underline will appear: 13 provides normal underlining, while lower values move the line upwards in the character cell to produce strike-over or overline effects. The EGA BIOS enables underlining whenever the adapter is switched to video mode 7 (monochrome), and disables it for the color modes.

It is easy to write a program to reenable underlining by setting the Underline Location Register; however, most software resets the video mode when it loads, thus nullifying any changes made to the EGA registers. The solution is to install a terminate-and-stay-resident (TSR) program that intercepts all the video mode setting calls and applies the fix after the mode is changed. An example is reproduced in UNDERLIN.ASM (listing 1).

The INTHAND procedure gains control on return from a mode-setting call (interrupt 10H, AH = 0). It enables the underline mode by writing the desired value (13 is used here) into the underline register. This register is loaded by writing its index (14H) to the EGA controller port, then writing the data value to the port at the next higher address. The base controller port address is available at location 40:63H of the BIOS data area.

Thereafter, any text written in colors 1 or 9 will be underlined. Although the attributes that produce underlining are fixed, the colors that correspond to those attributes are not. Therefore, the underlined colors may be changed by writing appropriate color values to palette registers 1 and 9. In this example, colors 2 (green) and 3AH (bright green) are used. Blue will no longer be available; instead, green will be available in underlined and plain versions. The underline colors are controlled by the values equated to labels COLOR1 and COLOR9. If desired, code may be added to change the background to a color other than black by writing a nonzero color value to palette register 0.

Gerry Kaplan is a computer consultant and software developer.

LISTING 1: UNDRLIN.ASM

```
SCNLIN  EQU     13              ;Scan line for underline.
COLOR1  EQU     0201H           ;Set color 1 to green
COLOR9  EQU     3A09H           ;Set color 9 to brite green
CODESEG SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODESEG, DS:COSESEG
        ORG     100H
ENTRY:  JMP     LOADER          ;Go to loader routine
OLD10   DD      ?               ;Storage for old INT 10H address
INTHAND PROC    FAR             ;Beginning of INTERRUPT Interceptor
        AND     AH,AH           ;Is it Set Mode function?
        JZ      CALLB           ;If so, arrange to come back from BIOS
        JMP     OLD10           ;Else go to BIOS and do not come back
CALLB:  PUSHF                   ;Push flags to simulate interrupt
        CALL    OLD10           ;Call BIOS and return
        PUSH    AX
        PUSH    DX              ;Save Registers
        PUSH    DS
        MOV     AX,40H          ;Set up addressability to BIOS DATA
        MOV     DS,AX
        MOV     DX,DS:63H       ;Get CRTC base address in DX
        MOV     AL,014H         ;CRT Reg 14 is Underline Register
        OUT     DX,AL           ;Indicate change to Reg 14
        INC     DX              ;Now point DX to CRTC data register
        MOV     AL,SCNLIN       ;Load AL with desired scan line
        OUT     DX,AL           ;Send data to CRT controller
        MOV     AX,1000H        ;Set a palette register
        MOV     BX,COLOR1       ;Set underline, color 1
        INT     10H
        MOV     AX,1000H        ;Set a palette register
        MOV     BX,COLOR9       ;Set brite underline, color 9
        INT     10H
        POP     DS
        POP     DX              ;Restore Registers
        POP     AX
        IRET                    ;Return to original caller
INTHAND ENDP
MESSAGE DB      'UNDERLIN: By Gerald S. Kaplan',0DH,0AH,'$'
LOADER  PROC    NEAR                    ;Loads the INT 10H TSR
        MOV     AL,10H                  ;Get vector for INT 10H
        MOV     AH,35H                  ;DOS function for get vector
        INT     21H
        MOV     WORD PTR OLD10,BX       ;Save the address
        MOV     WORD PTR OLD10+2,ES
        MOV     DX,OFFSET INTHAND       ;Point to new INT 10H handler
        MOV     AH,25H                  ;DOS call to set vector
        MOV     AL,10H                  ;Which interrupt to set
        INT     21H
        MOV     AX,0003H                ;Reset mode to turn on
        INT     10H                     ;     underlining
        MOV     AH,9                    ;DOS Print Message function
        MOV     DX,OFFSET MESSAGE       ;Point DX to message
        INT     21H                     ;Go and print message
        INT     27H                     ;Exit and stay resident
LOADER  ENDP                            ;(DX already points to
CODESEG ENDS                            :end of resident mode)
        END     ENTRY
```

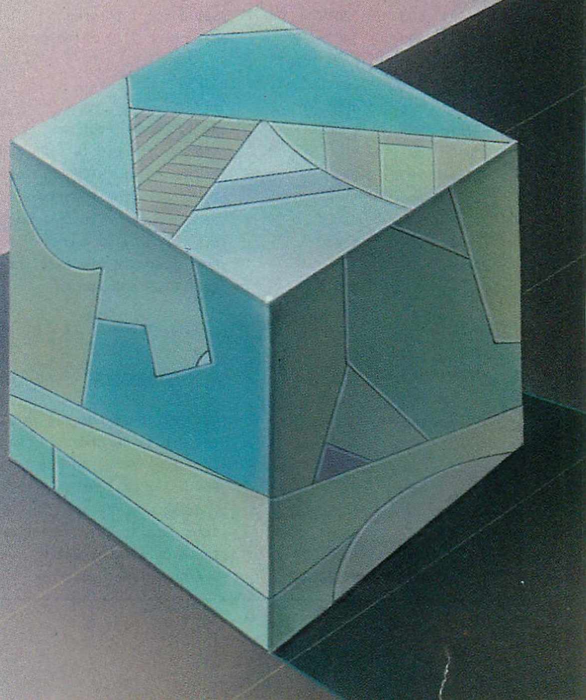
Tracking Code Modules

JIM VALLINO

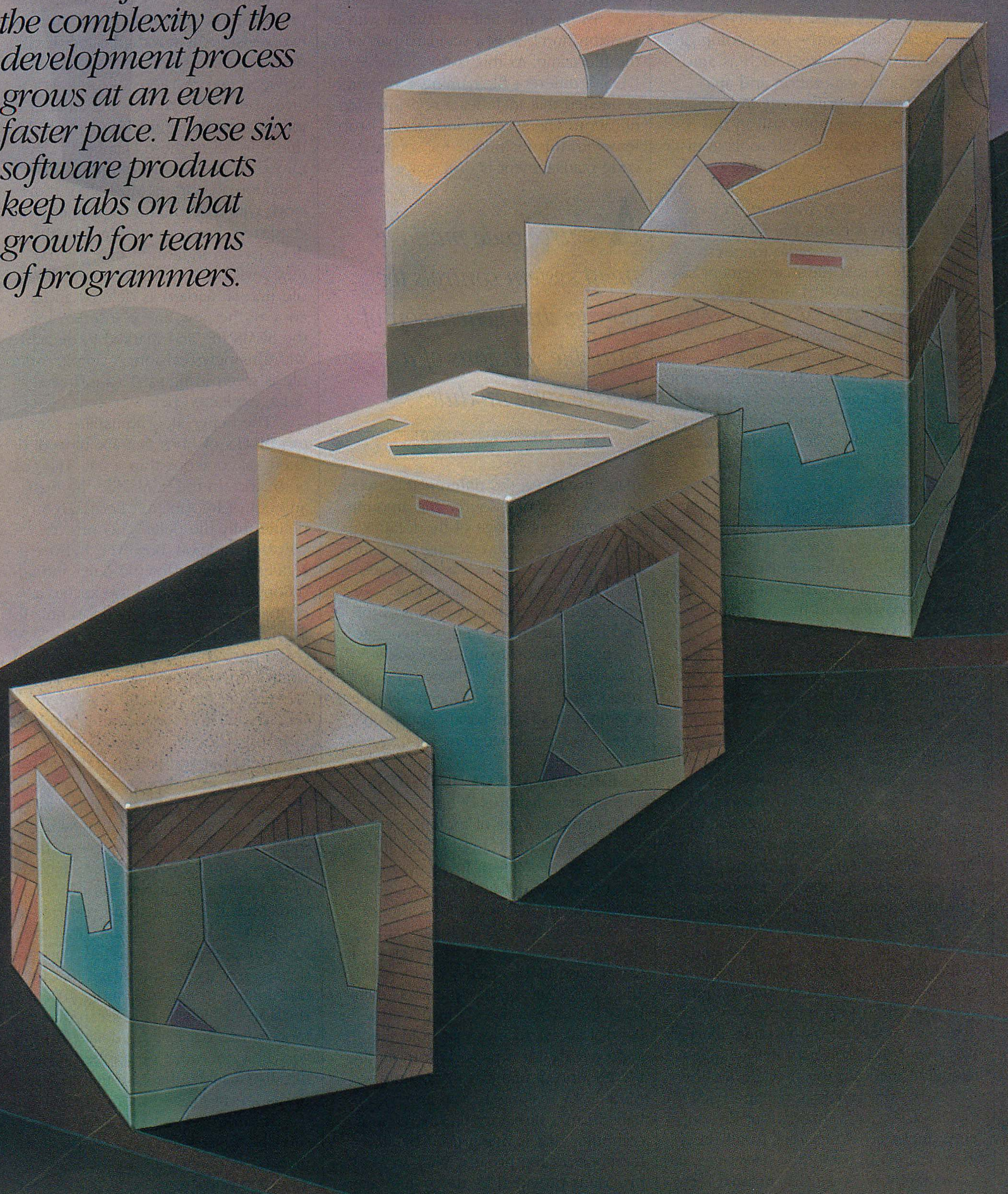
Today's feature-laden programs require an increasing amount of source code. With available memory no longer a problem for many microcomputer systems, PC programs have expanded to rival some mainframe entries in both size and complexity. Add to this the dynamics of the software market, where maintaining an edge means development cycles are shortened and more coding is performed in parallel by several programmers, and the fact that many programs are written to work on several different computer systems or different versions of an operating system. This state of affairs creates nightmares for many programmers managing source code.

When a program is composed of dozens of modules, keeping track manually of which set of source code files is the correct one to use for a given version of the program is tedious at best. The rush of last-minute coding before a release may cause the current information not to be distributed to all members of the programming staff, and time is wasted or programs built incorrectly are released. Considered here are six source code management systems (SCMS)—automated tools for controlling source code and changes to source code modules: (TLIB) by Burton Systems Software, BRICKS by Datalight, Source Code Interactive Librarian (SCIL) by H & S Associates, Corporate (PVCS)

ILLUSTRATION • ANDY LEVINE



As programs expand to incorporate more intricate features, the complexity of the development process grows at an even faster pace. These six software products keep tabs on that growth for teams of programmers.



by Polytron Corporation, (SRMS) by Quilt Computing, and (SVM) by Seidl Computer Engineering.

Tools such as these, that handle tasks normally performed by members of a software development team, aid in completing projects within tight time constraints. Such programs are classified as computer-assisted software engineering (CASE) tools. Currently, a great deal of development work is going on with CASE tools in areas such as automatic program generation and automated program verification and validation. Although generally still in the research phases, some CASE tools available for many years to software development teams on mainframes and minicomputers are also available on personal systems running the DOS.

A more correct name for these tools is ASCII text file management system. Any type of ASCII file can be placed under the control of an SCMS. This could include program design documents, user manuals, project schedules, and so on. It is as important to control changes to these vital pieces of information as it is to control a project's source code. A few of the problems that can be alleviated by source code management are:

- A programmer makes modifications to a module not knowing that conflicting changes may be made simultaneously by someone else.
- Changes are made to a module that are not correct and the original revision must be recovered.
- In order to respond to a problem report from the field, a specific prior program release must be generated.
- A problem is discovered that was not present in a previous release of a program. The development staff wants to know what changes were made between the two releases.

A manual record-keeping system could be used to track much of the information necessary to eliminate these problems. For example, to eliminate simultaneous updates to a module, every time a programmer wished to modify a source code module, a log would need to be checked to verify that no one else "owned" the module. If no one did, the programmer would claim ownership of the module, alerting other team members that changes are being made to the module. Following a strict discipline of making backup copies of modules throughout development might provide access to all previous revisions of a module. However, it is frequently the case that manual techniques such as these break down because programmers find them tedious and eventually do not adhere to them.

In addition, the manual method of making a backup copy of the source file every time modifications are made to create a new revision has two immediate problems. The first is generating file names so that the different revisions can be identified. Some operating systems have this feature built-in with a revision number as a standard part of the file name. As this is not available in DOS, a different file-naming convention is needed that includes a revision identification. The second, and larger, problem with this technique for source code management is disk space. In many instances, the difference between two revisions of a source file amounts to a small percentage of the total size. Disk space is wasted storing the entire new file because much is duplicated from the previous revision. The changes between two revisions is the part necessary to be kept to generate the new revision from the previous one. These differences between two revisions of a file are collectively called a *delta*—this is the basic concept in source code management.

A source code management system controls the storage and generation of multiple revisions of a source code module.

KEEPING TRACK OF REVISIONS

The primary features of an SCMS are to control the storage of multiple revisions of a source code module and to provide a mechanism for generating any requested revision on demand. A development staff relieved of these burdens has more time to devote to design and coding, and is thereby able to increase its productivity.

SCMSs can be thought of as databases for program code in that they control access to files that hold the source code for a project. Without an SCMS, a programmer can obtain a module by making a copy of it, perhaps from a directory on one central computer that is being used by all members of the project. With an SCMS, programs provided by the system are used to generate a working copy of the requested source code. When modifications to a module must be made available to others on the project, SCMS programs again are used to incorporate the changes into the library archive file that holds information about all revisions of the module.

To compact storage, an SCMS stores changes to a source file in terms of deltas. The information common to two revisions is not duplicated. Each delta represents only the differences between two revisions of a file. For a given set of differences, the amount of information that must be saved to define fully all of these changes is determined by the sophistication of the delta detection algorithm. Most algorithms work with a resolution of a line. If any character changes within a line, the delta has an indication that, for the new revision, this line changed. The old line is marked as deleted and a copy of the new line is inserted into the SCMS file and marked to be active with this new revision. A simple example is given in figure 1 (a *forward* delta, explained below).

The better algorithms also can detect blocks of lines that are moved, deleted, or copied within a file. The goal is to store in the SCMS file the least amount of information necessary to define fully the delta. This process involves a trade-off, however, between reducing storage for the delta and additional time required to execute a complex delta detection algorithm. To further reduce storage space, the SCMS could perform a data-compaction algorithm on the information to be stored. Here again, the trade-off is between less storage and longer times required to insert or extract revisions of a file.

Delta generation can move in forward or reverse direction. Forward deltas begin with a full copy of the original file and each delta details what must be done in order to create the next revision. When the SCMS is asked to generate a given revision of a file, it starts with the original file contents and applies deltas one at a time until the revision desired is generated. Forward deltas carry the disadvantage that they take longer to generate more recent revisions, which usually are requested more often, because all deltas from the very first one must be applied to the file. It is usually the case that forward deltas are requested more often.

Reverse deltas work in the opposite direction. A full copy of the latest revision is maintained in the SCMS file and each delta represents changes for generating the previous revision of the

FIGURE 1: *File Revision Using a Delta*

BASE REVISION	
Line	**Identification**
1	Line 1 of the base file.
2	This line will be changed in the revision.
3	
4	The revised file will have two lines added before this.
5	
6	This line will be deleted in the revision.
7	The end of the base file.

DELTA	
C2,1	At line 2, delete 1 line and substitute 1 line.
	This is the revised second line.
I4,2	Before line 4 insert 2 lines.
	This line was inserted before line 4.
	This one also.
D6,1	Starting at line 6, delete 1 line.

SECOND REVISION	
Line	**Identification**
1	Line 1 of the base file.
2	This is the revised second line.
3	
4	This line was inserted before line 4.
5	This one also.
6	The revised file will have two lines added before this.
7	
8	The end of base file.

To be able to recreate either the base or the second revision, a source code management system (SCMS) must store only the base revision and the delta information. This requires less storage than would storing full copies of each revision.

source file. With this technique, creating the latest revision of a file is relatively quick and generating older revisions takes longer periods of time.

The process of retrieving a revision of a file is usually referred to as a *get* or *checkout* of the file. The SCMS generates a working copy of the desired revision from its archive files by applying the appropriate deltas to the base file. When the SCMS goes about the process of saving modifications to the file, differences between the modified file and the revision upon which it is based are detected and used to create the delta information. This operation is commonly called a *put*, *delta*, or *check-in* of the file.

Each revision of a file must have a unique identifier. One common method for assigning revision numbers is in the form **RM.Rm**. **RM** and **Rm** are numbers representing the major and minor revisions of a file respectively. The initial check-in of a file under a SCMS creates revision 1.0. Each subsequent get/put cycle will cause an increment to the minor revision number as the normal sequence. This would make the second revision 1.1. An example of a typical cycle of revisions for a file is shown in figure 2. When a major change occurs in a system, such as release of the product and beginning work on the next release, the SCMS is instructed to increment to the next major revision and reset the minor revision to 0. As would be expected, after the first major change of this type, files would be on revision 2.0.

This path of development, which is straight along the central line of major/minor revision numbers, is referred to as the *trunk* of the file. It is sometimes necessary to create parallel development paths for a single module. In the figure 2 example, this occurs with revision 1.1 and 1.2. The revisions of the file that diverge from the trunk are called *branches*. A branch might represent a path of development for a particular operating environment or customization provided for one client.

Branches generally are identified by adding a branch major and minor revision number to the trunk revision from which they diverge from. In the example, revision 1.2.1.1 is the first revision of a branch from the trunk revision 1.2. From this point new revisions on the branch increment the branch minor revision number. The example revision tree also shows a situation where a branch from revision 1.2 is merged back into the trunk at revision 3.0. Some SCMSs can perform these merges and generate error messages if a conflict occurs during the merge. A merge of this sort might be performed if the branch represented some customization for a particular client that now will be incorporated into the mainstream product.

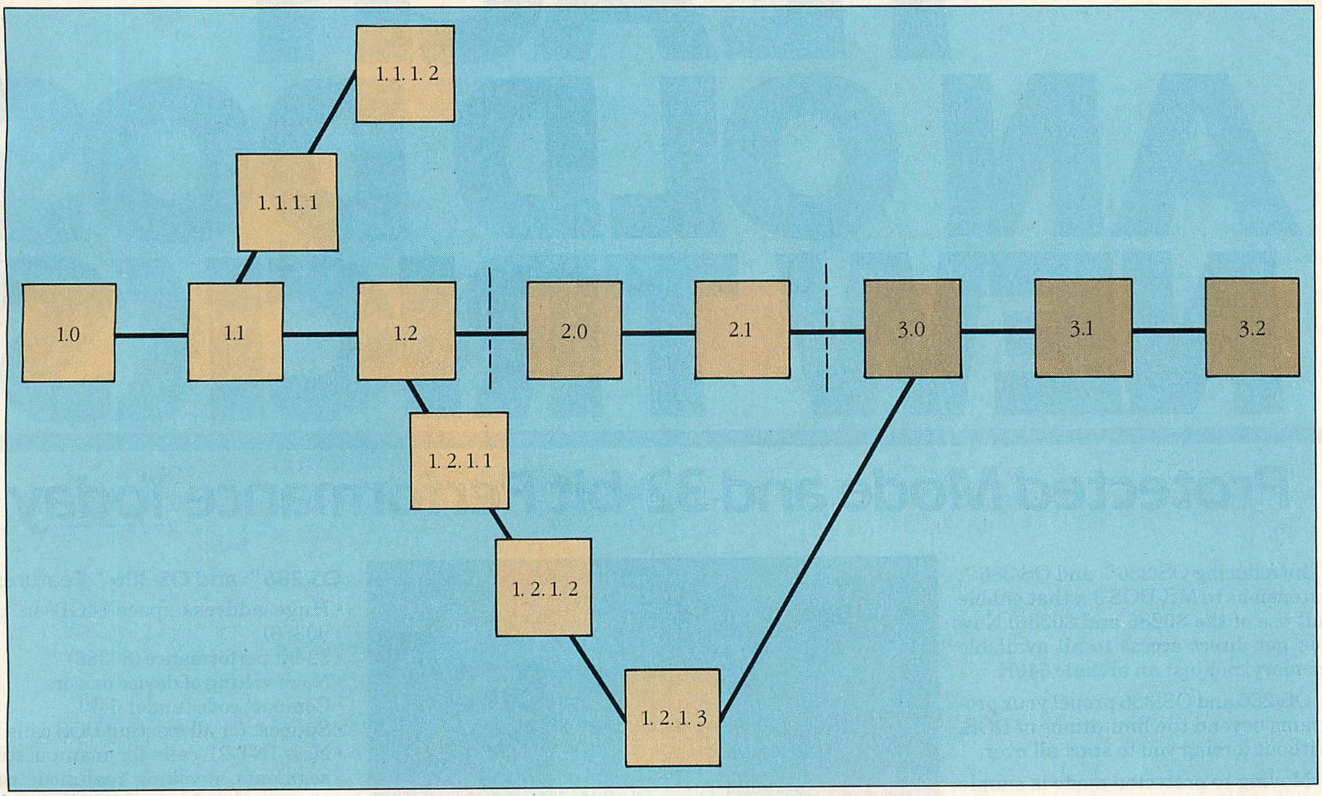
GETS, PUTS, AND LOCKS

When a revision of a file is needed, a get request is issued to the SCMS. All SCMS archive files for a given project are maintained in a single place, usually a directory on a file server. The exact structure of the files kept by the SCMS vary from one system to another. A system might place all information for a source file it manages into one file under its control. Another system uses several files and/or directories to maintain the information. (When backing up files a system that uses a single file for all information might be considered to have a slight edge since only one file needs to be saved). Systems that require a specific directory structure are a bit more difficult to use than those in which archive files can be in any directory. This is particularly true if source files are used in several projects, each of which is developed in a separate directory. To prevent accidental deletion, it is best if the SCMS can set the DOS read-only file attribute on all of these files.

Performing a get without specifying an explicit revision number usually defaults to the latest revision on the trunk. The latest revision on the main trunk or a branch is referred to as the tip revision. If a revision is specified, then the SCMS will generate that specific revision number. The result of the get is a working copy of the source file as it existed when it was checked in to create the revision requested. This retrieved working copy might represent how the file looked yesterday afternoon or six months ago. The SCMS maintains all intervening revisions of the program and sorts out changes.

The SCMS get provides a mechanism for one programmer to check out a revision of a file for editing. This places a lock on the revision and prevents two programmers from simultaneously making changes to the same revision of a file. Anyone else who attempts to check out that revision for modification would be alerted that the

FIGURE 2: *Revisions in a Single File*



Two revision branches that start with revision 1.1 and 1.2 are shown. The last revision or branch on the trunk is referred to as the tip. Branches also can merge back into the trunk, as is shown for the branch from revision 1.2 (1.2.1.3).

revision is locked. A read-only copy, needed for viewing or compiling the file, can always be obtained from the SCMS. No lock is placed on the file when a user performs this operation and if the SCMS sets the DOS read-only file attribute on the copy of the revision generated, the file is more difficult to modify and thus reinforces the read-only notion. A revision that is retrieved for editing would have its read-only attribute cleared.

If, when creating a working copy of a revision, the SCMS finds that a writable copy of the same file already exists in the user's directory, it should provide an indication of this before overwriting it. This check prevents losing valuable modifications when a programmer, who has a revision out for editing, wants to check something in another revision of the same source file and requests a second copy of the file. The edit lock is removed from a revision when the owner checks it in with a delta operation or manually removes the lock.

Identifying the team member who "owns" the edit lock on the file is quite useful. Someone who needs to make modifications to a locked revision is made aware of who must check-in the file to make it able to be edited again. DOS has no user identification (ID) information available; SCMSs incorporate user ID information by means of environment variables or prompting for a user ID. It is also necessary for an SCMS that supports branches to allow more than one revision of a source file to be edit-locked at the same time. This allows modifications to be performed on several branches of the file in parallel. The SCMS should track which revisions are locked and by whom.

When a file that has been modified is checked in, the SCMS must be able to determine the revision from which to create the delta. If the system allows multiple revisions to have edit locks and employs a user ID, then the default is the revision that the user owns for editing. Otherwise, the default would be the single revision that is locked. For special circumstances, such as creating a new major revision, the user must have the capability to override the default revision number when making a delta. In general, it should be the case that any attempt to apply modifications to a revision that is not locked or that the user does not own generates an error.

SYMBOLIC VERSION LABELING

Many times during a project, current revisions of every module may be used to build a complete system. This result of this building might represent a release of the product or a version that is to undergo internal system testing. This frozen configuration of the system often is referred to as a *version* of the system. A mechanism for remembering the source module configuration used to generate a version is needed so that it can be recreated upon request at any time in the future.

The symbolic version labeling feature provided with many SCMSs gives the source code librarian the capability to associate a symbolic version label with specific revisions of the source files. Figure 3 shows an example of a small program composed of four source files. Each of these files has reached its current state by going through several revisions. This is represented vertically under each of the file names. The horizontal lines connecting revisions of each file represent the source code configuration at the time a version of the program was created. These versions have the symbolic labels Release_1, Release_2, System_test, and Development.

TEACH AN OLD DOS NEW TRICKS

Protected Mode and 32-bit Performance Today

Introducing OS/286™ and OS/386™, extensions to MS-DOS 3.x that enable full use of the 80286 and 80386. Now you get direct access to all available memory, not just an archaic 640K.

OS/286 and OS/386 propel your programs beyond the limitations of DOS, without forcing you to start all over.

Moving to protected mode is simple because OS/286 and OS/386 give you *the same interface as DOS*. The hardware is still under your direct control, many 16-bit compilers already generate code suitable for OS/286 and OS/386, and existing highly tuned, machine-specific subroutines running in real mode can be efficiently called from within protected mode. Since most of your own code won't need to be rewritten, your programming investment is preserved. And because OS/286 and OS/386 work *with* DOS 3.x, they don't affect other programs, device drivers, or TSRs.

In addition to the larger address space offered by protected mode, OS/386 adds 32-bit performance to systems like the Compaq™ 386 which, until now, have been shackled to 8086 emulation.

Dhrystone Benchmarks	Microsoft C 16-bit	High C OS/386 32-bit	Scale
IBM AT 6Mhz	793	na	1.0
Compaq 386-16Mhz	2,380	5,837	7.3
HummingBoard-16Mhz	2,777	6,718	8.5
HummingBoard-20Mhz	3,571	8,470	10.7
Vax 8600 (Unix 4.3 BSD,cc)		6,423	8.1
Sun 3/160 (Sun 4.2 3.0A,cc)		3,246	4.1

OS/386 can be customized to give <u>unmodified</u> DOS programs up to 900k on 386 systems, regardless of how many TSR's, networks, disk caches, etc., are installed.

OS/286™ and OS/386™ Features:

- Huge address space (4GB on the 80386)
- 32-bit performance (80386)
- No rewriting of device drivers
- Compact code (under 64k)
- Support for all existing DOS calls
- New INT-21 calls for manipulating segments, invoking real-mode routines and interrupt handlers, and addressing physical memory directly
- Full interrupt vector support
- Powerful debugging: concurrent DOS environment while debugging protected mode programs
- The ability to run non-Windows programs in a window

A.I. Architects gives you a complete development toolkit:

OS/286 or OS/386 kernel and linker, Symbolic debugger and command processor

Options include:

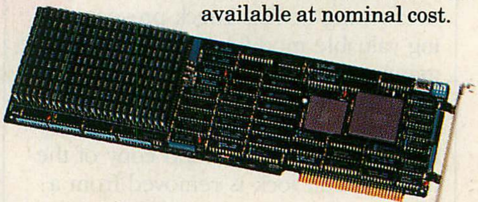
16-bit and 32-bit compilers

High C, Professional Pascal, or F77L FORTRAN

32-bit Assembler

386 HummingBoard™

The basic Developer's Kit is $495. 32-bit Compilers are $895. Run time licenses for OS/286 and OS/386 are available at nominal cost.



A.I. Architect's HummingBoard™ is a high performance 386 coprocessor for the PC-XT, AT and compatibles available with the 80387 and 2-24 Mbytes of RAM.

A.I. Architects, Inc.
One Kendall Square
Cambridge, Massachusetts 02139
(617) 577-8052

OS/286, OS/386 and HummingBoard are trademarks of A.I. Architects, Inc., Compaq Deskpro 386 is a trademark of Compaq Computer Corp., High C and Professional Pascal are trademarks of Metaware, Inc., F77L FORTRAN is a trademark of Lahey Computer Systems, Inc., Microsoft and MS-DOS are trademarks of Microsoft Corp., VAX 8600 is a trademark of Digital Equipment Corp., Sun 3/160 is a trademark of Sun Microsystems, Inc., Unix is a trademark of AT&T.

CIRCLE NO. 253 ON READER SERVICE CARD

Release_1 consists of File 1, revision 1.1; File 2, revision 1.0; and File 3, revision 1.2. Note that after Release_1 was made, all files were changed to a new major revision number representing the work on Release_2. Often the first revision of a new major release number is created by getting out the last revision of a file and putting it back unchanged at the new major revision number. This process was repeated after Release_2 was created.

The current work on the next release has two symbolic labels associated with it. One represents the version that is undergoing to system test. The System_test version is composed of revisions of File 1, File 2, File 3, and a revision of the new File 4. As work on a project progresses, individual programmers complete work on program features that are needed by other team members. The Development version represents the revision of each file that the responsible programmer wants other team members to use. Revisions of a file beyond the one with the Development version label represent continued development that the programmer does not believe to be sufficiently stable for use by other programmers.

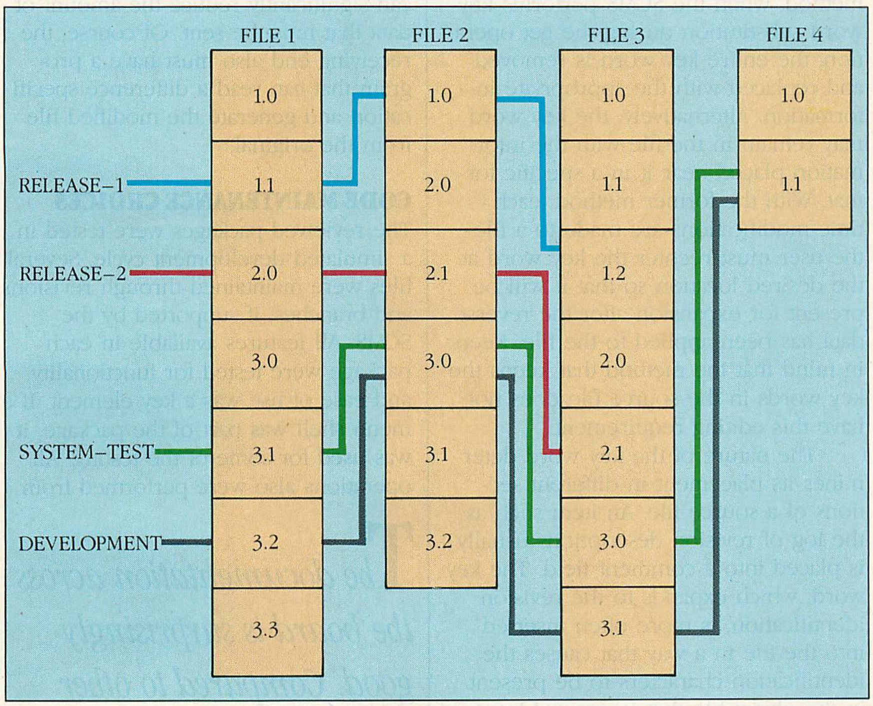
When a project member wants to build an executable system for test purposes, he would get copies of the revision of all source files carrying the Development symbolic version label. If the programmer is testing new features, then work files may be present that have modifications beyond the Development revision of that file. The SCMS should not overwrite these modified files when the user tries to get the Development revision of the file. If the DOS read-only file attribute is used to identify files that the SCMS considers to be nonmodifiable, then it is an easy matter to generate a warning whenever the SCMS attempts to overwrite a file that is not read-only. Alternatively, the file's time/date stamp can indicate a file that has been modified.

An SCMS that supports version labels should provide easy mechanisms for a label to be associated with a revision of a file, disassociated with it, or changed to another revision. Also, to create the opportunity to build a specific version of a program at a future time, it should provide a method for retrieving the revisions of all source files associated with a version label.

HISTORY AND JOURNALING

Many times project staff members need to know the difference between two revisions of a source file, either at the

FIGURE 3: *Revisions of Several Related Files*



Files may be connected to a symbolic version label. Source file revisions are shown vertically; symbolic version label associations are the horizontal lines.

detailed source code level or in terms of a summary description of the changes made. The auxiliary programs discussed below provide the capability to show which lines of source code have changed. SCMSs provide a capability to annotate the deltas made to a file. Whenever a file is initially put under code maintenance or when a revision is checked in, the user is prompted for a comment. This description of the revision becomes a permanent part of the history for the file. Some systems limit the size of these comments to a single line, while others allow multiline revision descriptions. Reviewing these descriptions is useful when a problem is detected in an unexpected area and clues are needed to identify suspect modules.

An important element to managing software development is tracking changes to the software. The features of an SCMS discussed so far assist in controlling changes to the source code. An SCMS that provides reports of activity is useful for a project manager who wants to follow development activity. Some systems maintain a journal of all actions performed on the files under SCMS control. This journal would record, for example, all get or put operations along with file name, revision, date/time and user ID. A project manager can use this to track development activity or a code librarian may get useful information from the journal file if a problem develops in the SCMS library.

KEY WORDS

When a revision of a file is called from the SCMS, the user might want to have some revision identifying information contained in the source file. This information might include revision number, date, programmer name, and/or reason for modification. If the programmer follows a strict discipline, the information could be placed into the source file whenever modifications are made. However, all of this information is maintained by the SCMS as it tracks changes to the source file and several systems provide a key word expansion mechanism to include the desired information in the work file created.

Key words such as Revision or Comments are placed into the source file in a defined format. When a get operation is executed, the appropriate data are inserted wherever the key words are found in the file. With keyword substitution, the working copy of a source file is easily identified as to its revision number, time/date stamp, or user ID of programmer creating the revision. If comments were included, then the revision descriptions for all previous modifications to the module are part of the work file.

Key-word substitution is performed in either one of two ways. In the first method, when the SCMS performs key-word substitution during the get operation, the entire key word is removed and replaced with the appropriate information. Alternatively, the key word may remain in the file with the information placed near it in a specific format. With the former method, each time modifications are made to a file the user must reenter the key word at the desired location so that it will be present for expansion after the reverse data has been applied to the file. Keep in mind that the method that keeps the key words in the source file does not have this editing requirement.

The nature of the key word determines its placement in different sections of a source file. An item such as the log of revision descriptions usually is placed into a comment field. The key word, which expands to the revision identification, is more often inserted into the file in a way that causes the identification characters to be present in the object file that is derived from this source file. One technique for accomplishing this is to place the key word inside a string definition. This module identification then also would be contained in the final executable program file. If the revision information is preserved, then a utility to display any strings of characters found in a file can scan an object file or executable file to identify the source configuration that created it.

An SCMS is composed of one or several programs that perform its various tasks. Some systems also include a surrounding shell to provide a more friendly interface to the system. Most SCMSs also have auxiliary programs or operations that aid in source code management. These extras provide functions beyond the routine get and put operations.

One of the more common auxiliary programs is a *difference analyzer*. Such a program performs the same delta detection algorithm that is used when modified files are checked in and displays the difference between two files or between a work file and a revision under source control. A display of these differences can be quite useful if some time has lapsed since the last modification was performed on a file and the programmer needs to review what modifications already have been completed or to determine exactly what changes were made between a working and nonworking revision of a module. If changes to source code need to be delivered, especially via telephone lines, a difference program can significantly reduce the amount of data that must be sent. Of course, the receiving end also must have a program that can read a difference specification and generate the modified file from the original.

CODE MAINTENANCE CHOICES

The reviewed packages were tested in a simulated development cycle. Several files were maintained through revisions and branches, if supported by the SCMS. All features available in each package were tested for functionality, and ease of use was a key element. If a menu shell was part of the package, it was used for some of the testing. All operations also were performed from the command line to ensure that operation from a batch file or automatic program builder is possible (see the accompanying sidebar, "Automatic Program Building"). Table 1 lists the features of the products reviewed.

The documentation across the board is surprisingly good. Compared to other software products, it was easy to locate information.

The test machine was an IBM PC running at 4.77 MHZ with 640KB of memory and a Maynard Electronics 30MB Apollo disk installed. To minimize disk access differences, the hard disk held only those files necessary for the specific test being run. In all, three tests were devised to demonstrate the relative performance of each package.

Test 1 is a file that started as a single line of characters and was placed under source code management. The file then was repeatedly taken out for edit, with a single line added at the end of the file, then the file was checked in with a short comment. This process was repeated 100 times; the sizes of the resulting archive files for each SCMS are shown in table 2, along with the time it took to get the latest revision. Also shown is the amount of time required to retrieve the first revision of the file after 100 deltas.

The second test is a 50-line C program, a file that was put under control of the SCMS, then taken out for edit. Each line in the file was transposed with the next line in the file. Resulting sizes and times for revision extraction are shown in table 2.

Test 3 is a text file with 47,073 characters on 1,473 lines. This file was placed under source control and then retrieved for editing. A single line near the end of the file was modified by the addition of 21 characters to the middle of it, then it was checked in. Both the size of the archive file and the time to perform the delta operation for each system are shown table 2.

The documentation across the board is surprisingly good. Compared to the material distributed with other software products, it was easy to locate information. All but one of the manuals are in the standard 5.5-by-8.5 inch size, with typeset pages, and each manual is distributed in a binder. None of them contained errors or misstatements that would have prevented the correct operation of the software. (Additional comments are made in the review of each product.)

Burton Systems Software. TLIB checks in as the least expensive product reviewed, yet it is packed with features. The programs that make up the system are distributed in a **.arc** file. To use the programs, they first must be extracted using a provided archive utility. (The archive program, PKXARC, is not completely compatible with the ARC program used by PCTECHline.) It also has an archive of several public domain programs. Included in this collection is a MAKE-type program complete with source code, two keyboard enhancers, and several other utility programs.

A single executable file performs all of the source management operations for TLIB. When the program starts, the current directory is searched for a configuration file called **tlib.cfg**. If found, it is read to configure the program. If this file does not exist, the environment is searched for a variable named **tlibcfg**, which, if found, is used to specify the name of a configuration file. Configuration files cannot be nested, and no means is offered of specifying a configuration file on the command line. Several modes of operation can be enabled and disabled from the configuration file.

TLIB is the only system reviewed that, by default, can create a modification to the text in a source file. TLIB has an unacceptable default mode for handling tabs: upon check-in of a file and whenever possible, it converts all sequential spaces to tabs, assumed to be at eight-column separations. Upon

check-out, tabs are converted back to spaces, again assuming tabs every eight columns. But many files are edited with a four-column, tab-stop setting, and the TLIB default still will retrieve the file with spaces inserted for tabs assuming tab stops at every eight columns. Fortunately, TLIB has a configuration option to leave tabs and spaces as they are during both check in and check out. Another file modification that is performed—and cannot be optionally turned off—is the stripping of trailing blanks on a line. If an application requires that these spaces remain, then TLIB cannot be used.

The archive files maintained by TLIB can be placed in any directory by specifying a PATH configuration option to TLIB. Part of this option also tells the system how to generate the name for the archive file based on the extension of the working file being stored. This extension calculation is not quite as flexible as that provided in Polytron's PVCS, but is much better than the default extension of .TLB. Using the default causes collisions between files such as **file.c** and **file.h** when both are put under source control. The calculation in TLIB permits combining characters from the source extension with specified characters to create a unique extension such as **.c$** for **.c** files and **.a$m** or **.asm**. A configuration option can be selected that causes archive files to be made read-only to minimize the possibility of accidental deletion.

Revisions of files are identified by a single integer number with support for neither major/minor revision numbering nor branches. Files can be locked when changes are to be made. The lock information is stored in a separate file in the archive directory. A calculation for the extension on this file also can be given in the PATH option. The lock information includes a user ID obtained from the an environment string TLIBID, or in the configuration file ID option. User ID is checked when the file is returned after modification, and the check-in is aborted if they do not match. Locking must be enabled in the configuration file and the necessary commands also must be enabled. When a file is checked out for read-only use, referred to as *browsing*, the work file does not have its read-only file attribute set.

Some problems were found with the operation of the K command. This command checks in a file that had been checked out for editing, then immediately checks it out for editing again. When an unmodified file was checked in with the K command, the operation aborted because the FORCEU option was not selected to allow check-

AUTOMATIC PROGRAM BUILDING

Source code management is one of several steps involved in getting a program into an executable form. The SCMS can take care of making sure that all of the proper revisions of the source files are present. Then, many steps must be followed in sequence to create the final program. These steps often are specified in a way that allows for an automatic program building utility, such as MAKE, which is available under UNIX and with several compilers available for use with DOS, to rebuild the entire program.

In a make file, all of the files that must be present for the program to be built are declared. The dependencies of certain files on other files is also declared along with the steps necessary to derive a target file from its dependent files. A target file is considered up to date if it has a later time/date stamp than all of its dependent files. If any of the dependent files has a newer time/date stamp than the target, all steps specified in the make file are executed to generate a target file. If one of the dependent files is itself dependent on other files, then it is first rebuilt, if necessary, to bring it up to date.

For this process to work the time/date stamp on files is critical. The safest approach is to have a work file extracted from the SCMS archive with the current time and date. This would force the rebuilding of all of the files that are dependent on this file. When a previous version of an entire program is to be built, often the operation is performed in an empty directory beginning with a **get** of all the appropriate revisions of the source files for this program version. The entire system is then rebuilt.

For day-to-day development, operating in this manner can cause extra compilations. Storing object files in an object library and then deleting the object from the development directory is common practice. The time/date stamp on the library file is not a reliable indication of the time/date of one of the objects that it holds. A smarter program builder could determine the time/date of an object file within the library and then determine if a compilation must be done. Carrying this one step further, the program builder could deal entirely with revision IDs. Using an SCMS that includes revision identification that propagates into the object files, a program builder could detect if the object file was derived from the requested revision of the source file. The time/date stamps are ignored completely because they are an artifact of the real issue, namely, "Did this object come from a certain revision of the source file." Both Polytron's and Seidl's source code management systems have separate MAKE type products that provide some of these features.

At present, automatic program builders make an attempt to eliminate inconsistencies—that is, they make sure the object code that is used to build a working system was generated from the correct revisions of all the source files. The next generation of these programs will no doubt use artificial intelligence techniques to manage inconsistencies rather than completely eliminate them. Rules will specify permissible inconsistencies. One such example is if the object code dependent on a source file is not up-to-date, but the changes that occurred in the later revisions of the source file are only in comment fields. This type of inconsistency can remain without affecting the operation of the final program, and the recompilation step can be omitted.

AI-based program builders will have a tight coupling with the compilers and SCMS. Information will be available concerning the areas of a program that have been affected by changes in the source code. The automatic program builder will use this information to manage the inconsistency between the object file and source file and provide the user with details of that areas of the program that are not up-to-date. The user will specify rules that define the sections of a program and the allowable level of inconsistency. The result will be fewer operations performed to generate development copies of the executable programs. Having fully consistent final production versions of the programs released would be prudent.

—Jim Vallino

If you ever wanted to take a crack at assembly language, now's the time.

You probably already know that assembly language subroutines are the smartest way to get the fastest programs.

But if the complexities of working in assembler made you think twice, here's some good news. We've made Microsoft® Macro Assembler Version 5.0 a lot easier to use.

We eased the learning process by giving you the best support around. We completely revised our documentation. The new Mixed Language Programming Guide gives you step by step instructions for linking your assembly code with Microsoft QuickBASIC, C, FORTRAN, Pascal and other languages. And you get a comprehensive reference manual with listings of the instruction set and examples of each instruction. We didn't stop there, though. You also get an on-disk collection of templates and examples.

We've also dramatically simplified the high-level language interface. In just a few simple steps, you can be calling Macro Assembler subroutines from programs written in your favorite language.

Now that you're writing the fastest programs, Microsoft is giving you the fastest way to debug them. For the first time, we've added our CodeView® debugger to Macro Assembler.

With source code and comments on your screen, Microsoft CodeView makes debugging programs containing assembly language subroutines a snap.

And you'll be glad to know that you don't sacrifice any speed for all the ease of use. We took the fastest Macro Assembler on the market and made it even faster.

So what are you waiting for? Get your hands on Microsoft Macro Assembler and see what it's like to break your personal speed limit.

Microsoft®

For more information or for the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada, call (416) 673-7638.

CIRCLE NO. 150 ON READER SERVICE CARD

TABLE 1: *SCMS Features*

	BURTON	DATALIGHT	H&S ASSOC.	POLYTRON	QUILT	SEIDL
Product	TLIB	BRICKS	SCIL	PVCS	SRMS	SVM
Version	3.0	1.23	1.24	1.4a	3.0	1.05
Price	$99.95	$159.00	$349.00	$395.00	$185.00	$299.95
Specific directory structure	○	●	●[a]	○	○	●
Revision locking	●[b]	○	●	●[b]	●	○
User identification	●	○	●	●	○	○
Revision deletion	○	○	● (tip)	● (any)	● (tip)	○
Symbolic version labeling	Batch file	●	●	●	●	●
Branching	○	○	With vers.	Unlimited	1 level	8 levels
Merging	●	○	●	●	●	●
Key words	○[c]	○	○	●	Replace	○
Difference program	●	●	●	●	●	●
Activity logging	○	○	○	●	○	○
Screen-base interface	●	○	●	○	●	●
User configurable	●	○	○	●	●[d]	○
Archive file write-protected	●[b]	○	○	●[b]	●	●
Archive file name	Extension calculated	Work file	Fixed extension	Extension calculated	Work file	Work file
Revision comments	Single line	Multiline	Multiline	Multiline editable	Multiline	Multiline editable
Work file time/date stamp	Current	Current	Current	Original, current	Original, current	Original
Max. line length (char.)	254	512	300	65,535	255	512
Max. number of lines.	32,000	65,535	64,000	1 G lines	5,000	16,383
Max. number of revisions	32,767	65,535	999	1,500	500	>2 million
Manual size, style	110+ pp., typeset	40 pp., typeset	160+ pp., dot matrix	120 pp., typeset	130+ pp., typeset	215 pp., typeset

●=Yes
○=No

[a] *Current directory*
[b] *Optional*
[c] *Revision comments only*
[d] *Archive location*

All of the products reviewed in this article provide the basic functions that are necessary to code management, but two stand out: PVCS provides everything that is needed for a large project; TLIB is a very complete system for its low price.

ing in an unchanged file. The DELETESRC option was active, which caused the working file to be deleted. The lock on the file is kept, correctly, but no check-out of the file is performed. The file had to be checked out manually to get another working copy. The documentation says that DELETESRC does not affect the K command, but this does not seem to be true because the work file remained when a K command was performed with DELETESRC set negatively.

TLIB provides a unique feature among the products reviewed. The user can specify customized prompts and help messages in the configuration file. This allows for personalized operation or can provide information specific to maintaining code for a project.

A file that is checked in after editing can have a single comment line added as a revision description. As an option, TLIB can include in this line the check-in time and date. The accumulated revision history also can be added to the working file by key-word substitution. The specification of the location in the working file for this information is a little confusing at first, requiring column counting and line spacing information. No key word is provided for insertion of only the revision number as an identifier that would propagate through to object files and executable code. When a file is retrieved from the archive, no matter what revision is requested, the resultant working file will always have the current time and date. It must be said that this is a conservative approach for automatic program building since it forces a recompilation of the source file.

TLIB provides version labeling in a minimal fashion. The **tlibsnap** program creates a DOS batch file that has a **tlib** regress command for every file specified. The regress command specifies the latest revision as the one to retrieve from the archive. A single remark line gives a description of the revision retrieved. Files from several libraries can be placed in one snapshot file; new files can be appended later. This is an ASCII .BAT file, so any necessary changes can be made easily, and the files also can be maintained by the source code manager.

The documentation states that TLIB will support operation in a local area network (LAN) environment and recommends using the DOS 3.1 SHARE program. (However, the system was not tested on a network to validate this claim for purposes of this review.)

TLIB turned in an excellent performance on all three tests used for this review. On tests 1 and 2, it generated the smallest library files, although its times were average. When compared to the systems that do not perform a compaction technique when storing the data, it also generated the smallest file for test 3. It was, however, amazingly fast in generating the delta.

Datalight, Inc. BRICKS is a utilitarian SCMS—not many frills or options. The library of archive files under BRICKS

control are placed in a subdirectory called BRICKS, in the directory where the source files exist. This is a mandatory arrangement. The manual warns that the user should neither disturb any of the files in the BRICKS subdirectory, nor add new files to it. BRICKS archive files have the same name as the source file, which can lead to confusion when doing backups. The user can specify where temporary files are placed by setting an environment variable, TMP, to the name of a directory. A RAM disk can increase the speed of operation.

BRICKS has several separate programs to perform the SCMS functions. The two used most frequently are **bi** (BRICKS check in) and **bo** (BRICKS check out). The **bi** command is used to initially place a file under BRICKS. **Bi** creates a library file in the BRICKS subdirectory. (It also creates the BRICKS subdirectory if it does not exist.)

The get operation is performed using the **bo** command. Both **bo** and **bi** can have wild cards and lists of files specified for multiple-file "gets" or "puts." BRICKS keeps track of a file's status with a file time/date stamp. When a revision of a file is pulled, the work file gets the current time/date. The BRICKS library file's time/date is also modified to this value. No means is offered of user ID or for locking a revision when it is checked out. All checked-out files can be written to, as can the BRICKS libraries themselves. However, the lack of write-protect on the latter files could lead to disastrous results with an inadvertent deletion.

If a work file has a later time/date stamp than its corresponding BRICKS archive file, it is considered out of date. A **bo** operation will generate a warning and not overwrite an out-of-date file unless forced by a command line option. When the user performs a **bi**, the operation aborts if the file specified is up-to-date. The user can force a check in of the unchanged file via a command line option. The **bi** program checks time/date stamps only to determine if a file is up-to-date, which, in normal usage, is an accurate test. If the user wants to check in to a work file with a time/date stamp that is prior to the last **bo** operation on that archive file, the time/date stamp will have to be brought up to the current time.

Version labeling is provided by the **bthread** program. The current revision of every file specified is saved in a file. This thread file, as it is called by BRICKS, can be used later to check out these revisions of the files. The thread file is an ASCII file that can be edited to make changes to the version association for a single source file without changing any other associations.

TABLE 2: *SCMS Performance*

	BURTON	DATALIGHT	H&S ASSOC.	POLYTRON	QUILT	SEIDL
Product	TLIB	BRICKS	SCIL	PVCS	SRMS	SVM
Test 1 file size	5,760	8,274	41,449	9,785	11,566	9,901
Get 1.0	2.1	70.0	12.2	9.8	10.1	1.8
Get 1.100	4.8	4.8	11.4	6.0	9.8	1.8
Test 2 file size	1,599	2,245	2,709	1,785	2,405	1,844
Get 1.1	2.5	2.8	5.2	2.2	2.9	1.6
Get 1.2	2.7	1.9	5.4	2.2	3.0	1.5
Test 3 file size	47,198	47,334	49,741	47,370	47,282	46,198[a]
Put 1.2 (min:sec)	0:19	1:44	1:16	0:41	5:29	0:48

All file sizes are in bytes. All times are in seconds except where indicated.
[a] In normal file mode; when compression is used, file size is 30,518 and time required for put is 0:44.

PVCS, SVM, and TLIB all exhibited good performance in the three tests for this article. SVM generated a smaller file in test 3 than the others using its normal file mode and a 34-percent smaller file when using its file compression option.

When a file is checked in, the original source code remains in the development directory. Automatic deletion of the working file is not provided. BRICKS strips the trailing Ctrl-Z (DOS end-of-file, or EOF, marker) from a file. In one instance, when a file that started with an EOF terminator was checked in and then checked out, it came out one byte shorter minus the Ctrl-Z. Whether or not this is a problem will depend upon the editor used.

The **blog** program displays the revision history of one or several files. Multiline revision descriptions are entered by the user when the **bi** command is executed. Editing of a description can be done only on the current line being entered. If multiple files are checked in together, the user can, optionally, use the same revision description for all. A program also is provided to display differences between two files or a file and a revision in a BRICKS library archive. This information cannot be used to update a file because the package has no program to generate a new file from the original and a set of differences. Neither does BRICKS have facilities for deleting a revision from a library file. Branches and key-word substitution are not supported.

BRICKS' performance on the tests falls in the middle of the pack. It is obvious from the time differences for extracting the first and last revisions of the file in test 1 that BRICKS uses reverse deltas: its extraction time for the first revision was the longest.

H & S Associates. The Source Code Interactive Librarian (SCIL) is a complete SCMS package; even so, this implementation leaves a lot to be desired. Its philosophy is unconventional compared to that of the other products reviewed here, and much of its nomenclature is different from what has become common usage to other SCMSs.

SCIL runs from a single program using either an interactive mode, or the user can specify the necessary information on the command line. The interactive mode is line-oriented using a question-and-answer format. SCIL asks the questions and the user types the answers. Help information is available at the various prompts for input, but the help text files are required to exist in the current directory. This can be cumbersome especially in cases where several projects in separate directories each requires a copy of all help files.

SCIL uses the concept of a library of source files. The library contains several frames, each of which can correspond to one source file module in the program. Information about library configuration is kept in a .SCL file, and each frame is maintained in a .SCF file. None of these files is write-protected. Before a file can be put under source control by SCIL, its frame must be added to the library. As a frame is modified, it progresses through several "levels," each of which is equivalent to a revision. Level numbers are single values from 1 to 999. Symbolic version labels are called *releases* and operate in a way that is similar to version labels in other products.

SCIL also adds the concept of a version; the SCIL version is a mechanism to specify that the changes made to a file are valid only for a specific version of the file. Source files can be retrieved by the version name, and only changes stored as part of that version will ap-

pear in the resultant work file. The SCIL manual specifies that this feature can be used to maintain one frame for several different operating systems or to maintain several source files in one frame. The utility of the latter is questionable, although the former provides a capability similar to branching. Merging of versions is available.

SCIL provides minimal user customizing or configuration for placement of files. All library files and frame files must reside in the current directory. For every command, the name of the library and the name of a frame must be entered. When in interactive mode, it would aid the user tremendously if the previous library name could be used as a default value.

Frame names can be a maximum of eight characters (periods and colons are not allowed). To distinguish between a source file **test.c** and **test.h**, the extension must be embedded in the frame name, such as **testc** and **testh**, respectively. This will cause a problem if the base file name plus extension is more than eight characters. When a frame is pulled from a library, a single extension is used as the default for every frame in the library. The frames above would be named **testc.get** and **testh.get** as defaults. This default file name can be overridden with the -G option, but no means is available to specify the desired file name from the command line. This makes its use in a batch file or by an automated program builder difficult.

To avoid some of these problems, the programmer can create a separate library for each type of file used. One possible arrangement keeps all C programs in a library of C frames and defaults the get extension to **.c**. Similarly, C include files would be in another library that used **.h** as the default get file name extension. Frame names are the same as the file name without an extension in this method. Each library also must specify a frame file extension because collisions could occur if all libraries used the .SCF default. The SCIL user manual provides no guidance or examples for organizing project source files as described above. The unfortunate part of this is that more and more files other than the frame file must be saved to do a complete backup.

One library parameter that can be changed when a library is initially created is the control record character (the default is @). The manual clearly states that no file maintained in the library can have a line that begins with the control record character. This is the only SCMS that places a constraint on the contents of files being controlled. The control record character can be changed, but only before any frames are actually stored in the library. If a problem develops later with a frame starting with the control record character, no method is specified to correct the problem.

During a get operation, SCIL generated an error message because of a missing lead comma before the frame name on the command line. It then continued to process the request only halfway—putting a lock on the frame but not actually creating a copy of it. This lock had to be removed manually to get a copy of the frame. During another operation, SCIL prompted for frame names when building a new library. The user has to provide a correct input value. The operation could not be aborted without pressing Ctrl-C. This caused the opened library description file to be left behind with a length of zero. When an attempt was made to build the library again, SCIL complained about the presence of this file and aborted the operation.

> ***The Polytron Version Control System is oriented toward flexibility and ease of use. Each system file performs one operation.***

SCIL does maintain locks on files that are checked out for editing. It is possible to write to all checked-out files, and the system does prompt for permission to overwrite an existing copy of the file. A name for the owner of the lock is requested. The default mode for the get operation is a check out for editing. In order to get a read-only copy, the **-x** option must be specified. A file can be checked in only if a corresponding get is active. Following a delta operation, the default is to delete the working copy of the file but this can be overridden with **-k**.

One display command shows the current options for the library, metrics such as lines that are added/deleted for each level of one or several frames, and the log of comments entered when frames are checked in. These level comments can be multiple lines long with simple line editing capabilities available when entering the information during a put operation.

The documentation for SCIL comes in a spiral-bound, 8.5-by-11 inch book with pages printed on a dot-matrix printer. Some sections, notably the index, were especially difficult to read because they had been printed in a compressed mode.

The SCIL file generated for test 1 was larger than the next biggest file by more than a factor of 4. It is also the case that SCIL performed slower than average in all three tests.

Polytron Corporation. The Polytron Version Control System (PVCS) is one of several CASE tools available from this company. PVCS is oriented toward flexibility and ease of use. The system's operation can be controlled mostly through configuration files. The system is composed of several programs, each performing one operation. When a PVCS command is given, the system begins by checking the current directory for the **vcs.cfg** file. If it is not found, the command then looks for an environment variable called VCSCONFIG and uses the associated string as the name of a configuration file. Items that can be specified in the configuration file include user ID name, the directory in which library archive files are maintained, and the directory in which temporary work files should be placed. The user ID can be specified in the configuration file or via an environment variable called VCSID. Configuration files can be nested, a feature that allows for project-wide configurations to automatically perform a user-specific configuration.

As a default, all files maintained by PVCS have their read-only file attribute set. This helps to prevent accidental deletion of library archive files. All the information for a given source file is kept in a single file maintained by PVCS. This file, referred to as the log file, is given the same base name as the source file and the extension is calculated using the LOGSUFFIX configuration option. For example, with the default value for LOGSUFFIX, PVCS keeps a **.c** work file in a **.c_v** log file, while .ASM files have a .ASV file extension.

Any operation that causes a change to a log file generates an entry in a journal file if the JOURNAL configuration option has been specified. Different projects or subsections of a large project can use different journal files by changing the name specified in the configuration file. A journal entry shows the log file, user ID, time/date and action performed. The VJOURNAL

program allows a user to display journal entries qualifying the selection in the following way: date, logfile name, operation, and user ID.

When a user checks a file out for editing, the revision can be locked. Multiple revisions of the same file can be locked by different users. A given user can, however, lock only one revision of a file. The locking information is maintained within the log file; thus, PVCS is the only system reviewed that does not use a separate file to support revision locks. An error message is generated if a user attempts to check out a locked revision for editing. When the revision is taken out for editing, the resultant work file is writable, while revisions that are checked out for reading have the MS-DOS read-only file attribute set. This can be changed with a configuration option so that all files checked out are writable. If a writable work file of the same name already exists when the checkout is performed, the user is asked for permission to overwrite. Command-line options for the GET command can specify automatic overwrite permission or denial that allows GET to run in a batch file. The extracted work file receives the revision's original time/date stamp by default or, if desired, it is possible to get the current time and date.

When a user checks in a revision, the system examines the user ID for a match with one of the revision locks. If the current user does not own a lock, an error message is generated. Otherwise, the delta is performed according to the revision information in the user's lock. All revision locking can be disabled from the configuration file but has an effect only when an archive is first created. After that, the locking action is stored in the log file and can only be changed using the VCS utility. If a revision other than the tip of a branch or the trunk is accessed for editing, PVCS assumes that a new branch is to be created from the revision that has been retrieved and will do so when the modified file is checked in.

PVCS provides one additional control feature that is not found on any of the other products in this review. A given source file can have an access list specified. The user ID of the operator who initially checked in the source file to PVCS is considered to be the owner of the file. This user may specify a list of user IDs to be considered the access list. Only user IDs on this list are given permission to access information in the log file. In this system, access lists are stored in the individual log files. The general VCS utility can be used to change the access list or owner ID.

PVCS allows revisions to be deleted. Most systems that have this capability allow revisions to be deleted only one at a time from the tip. In one operation, PVCS allows one or several revisions to be deleted from the tip. It is the only system reviewed that also allows revisions to be deleted from the middle of a development trunk or branch. The remove delta function, performed with the VCS utility, compensates for the removal of intermediate deltas by adjusting the remaining delta information. Caution should be exercised when deleting revisions using wild-card specifiers: PVCS generates an error message and aborts if the user tries to delete a locked revision or a revision with a branch.

The PVCS key-word expansion is the best, with key words for user ID, revision date, log file path, and revision description comments.

When a file is checked in, the user is prompted for revision comments. If an editor has been defined to PVCS in the configuration file, it is used for entering the comments. This takes a little longer to start the editor running, but it has the advantage of making it easier for the user to correctly enter multiline comments for the revision. If no editor is defined by the user, then a line editor with a full selection of commands is provided by PVCS.

The PVCS key-word expansion is the best of all the systems reviewed. Key words are available for user ID, revision date, log file path, and a log of all of the revision description comments. Key words are expanded by default. The expansion is not a substitution, however, which means that once a key word has been placed into a source file, it will not need to be inserted again; and for every get from PVCS, the key word will be expanded. This feature is useful in ensuring that revision identification always propagates through to object files and executable code. PVCS includes the **ident** program, which scans any type of file and prints out all key-word strings found in the file. This often is used to scan object and executable files to determine the source code configuration that created them.

The **vdiff** program calculates differences between revisions. Comparisons can be made between two work files, a work file and a revision in the log file, or two revisions in the same or different log files. The specification of the two sources for comparison is very flexible. The output from **vdiff** can be saved into a file to be used later by the **regen** program to create the second file from the original one. In addition, the merge program, **vmrg**, combines two source files into one.

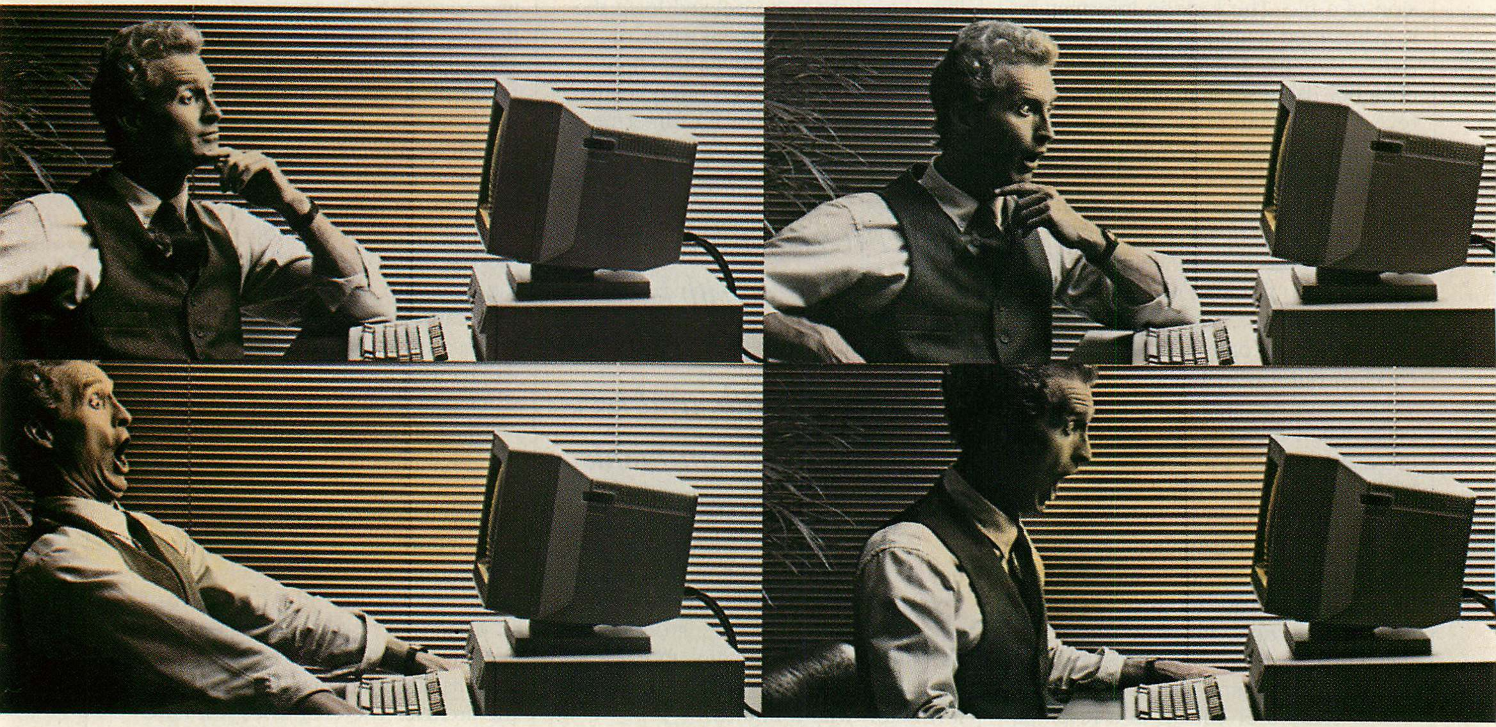
PVCS supports symbolic version labels. All label information is kept in the log files for each source file. As is the case with most PVCS commands, wild cards can be used to specify file names; this method provides an easy mechanism to label all the latest revisions of a project with a version association. If an association for the label specified already exists in a file, the user is prompted for permission to change it. Alternatively, a command line option can be set to grant or deny this permission automatically. Version labels can be used in PVCS commands in place of a revision specification.

PVCS also provides **vlog**, a program that displays information about files in the archive. The user does not have control over the type of information displayed. Output is either brief, which shows information only about the log file, or full, which shows file names, locks in place, number of revisions, and the description of each revision. Output can be qualified by date, revision, version, or author.

The first version of PVCS tested for this review (1.3) exhibited some unexpected behavior when used with certain files. The delta detection algorithm had difficulty with files that contained few unique lines. Small changes in such a file resulted in large changes in the size of the archive file; however, such files are atypical in normal use. The problem was observed during this review when a large file was created by duplicating a smaller file many times. This problem did not occur with any of the files that were generated through a "normal" development process. When notified, Polytron support personnel provided an updated version of PVCS (1.4a) that solves the problem. Users who encounter this problem also may obtain an update. PVCS performed well in the benchmark tests.

The Polytron documentation provides a good description of how librar-

EXTRA EXCITEMENT FOR THE NEW GENERATION.



INTRODUCING THE MOST POWERFUL 3270 SOFTWARE FOR THE PS/2.™

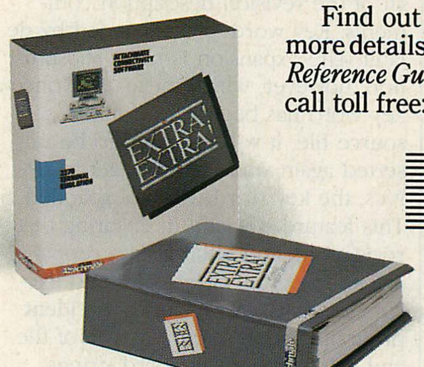
If you think EXTRA! Connectivity Software™ is just another dumb terminal emulator, hold on to your keyboard because you're in for a powerful surprise.

EXTRA! offers more connections to high places for the new generation of IBM® PS/2™ computers and the PC/XT/AT—even the 3270-PC. EXTRA!'s universal connectivity lets you keep the same friendly software over coax, modems, or local area networks using hardware adapters from IBM, Attachmate or IRMA.™

Extra Power at Your Fingertips.

EXTRA!'s powerful features include up to four mainframe sessions, printer emulation, and a choice of file transfers—including a better way to exchange documents and PC files with DISOSS. Plus the power of our IBM compatible 3270-PC API and HLLAPI.

You even have the options of adding host graphics and LAN Gateways.



Find out what all the excitement's about. For more details on EXTRA! and a free copy of our *Quick Reference Guide for Micro-Mainframe Communications,* call toll free:

1-800-426-6283

Attachmate

Quality Micro-Mainframe Solutions.

Attachmate Corporation
3241 118th S.E., Bellevue, WA 98005
(206) 644-4010

CIRCLE NO. 165 ON READER SERVICE CARD

ies might be set up for a single programmer shop, multiprogrammer nonnetworked environment, or a networked environment. It also offers a clear explanation of exactly how the networked operation works, including the mechanism that is used for locking log files from simultaneous updates. PVCS does not require the DOS SHARE program in order to support networked operation. (However, no test of networked operation were performed for this review.)

Polytron currently offers two versions of PVCS, network and corporate. Network:PVCS supports operation in a networked environment; Corporate:PVCS runs on one machine in an environment that is not networked. Both products support the full set of features. Polytron also has a third version called Personal:PVCS, which is meant for a single programmer operation and does not support user IDs or revision locking. Some of the auxiliary programs, such as **vdiff** and **vmrg**, are not supplied in the latter version.

Quilt Computing. Quilt's Software Revision Management System (SRMS) is the easiest package to use, compared to the other of these systems that provide a screen-based menu. SRMS is composed of several executable files, each of which performs one of the operations for managing source code. The initial menu screen displays a list of most of the programs that can be run. Once a program is selected, a second screen appears with entries for options and file names. In the lower half of the screen is a list of all available options and a short description of each. An error message is generated if the user attempts to run an operation without providing all necessary information. The menu displays a particularly good use of color highlighting.

Archive files under the control of SRMS can be kept in any directory. The environment variable SRMS$LIB identifies the directory in which the SRMS library archives are kept. Library archive files cannot be maintained in the working directory because they are given the same name as the source file. Instead, SRMS keeps information about the library in several files. These include a file named **srms.cfg**, over which the user has no control, the individual archive library files, and separate files to maintain locking information and symbolic version data. However, only the library files and the lock files have the read-only attribute set.

SRMS does not support a user ID but does determine the name of the current directory when the get operation retrieves a file for editing. That name is stored in the lock file for the source file retrieved. Each programmer on a given project can have his own directory to provide an indication of the owner of the lock. When working with SRMS, the programmer would change the current directory to his personal directory and then perform the necessary SCMS operations. Also, comments can be stored in the lock file that can include a user ID. When a delta is performed and the current directory does not match that stored in the lock file, the operation is aborted.

SRMS has some problems with its file locking. The most major of these is that a file is locked with an auxiliary file, the name of which is composed of the base name of the file being locked and a fixed extension of **.px**. If **module.c** is retrieved with locking, a file named **module.px** is created in the SRMS library directory. Then, when the user attempts to get **module.h** with locking, the system says that the file is already locked because **module.px** exists. The only way around this is to keep files that would collide in this manner in separate directories. Incidentally, the documentation makes no mention of this potential problem.

Quilt's Software Revision Management System is the easiest to use of those with screen-based menus; it clearly displays all options.

Another problem is that the **.px** file maintains information about the base revision and new revision that will be created when a file is checked in. The delta is performed from the base revision specified in the **.px** file. In many cases, this can lead to confusion. Assume a file is checked with locking, for example, revision 1.6 to create revision 1.7. When the file is checked in, the user wishes to create a new major release by specifying the delta as revision 2.0. The SRMS manual says that the next revision information in the **.px** file is ignored and the **.px** file is not deleted when the operation completes. The manual goes on to suggest that this file be manually deleted. If it is not, and someone checks in another revision (SRMS allows this though should not since no revision is legally out for edit) the system will create a new delta of 1.7 using a base of revision 1.6 as specified in the **.px** file. Now a new revision is present on the trunk that is located numerically between revision 1.6 and 2.0. When the next get is performed, however, revision 1.7 is considered the tip and it is checked out to create 1.8.

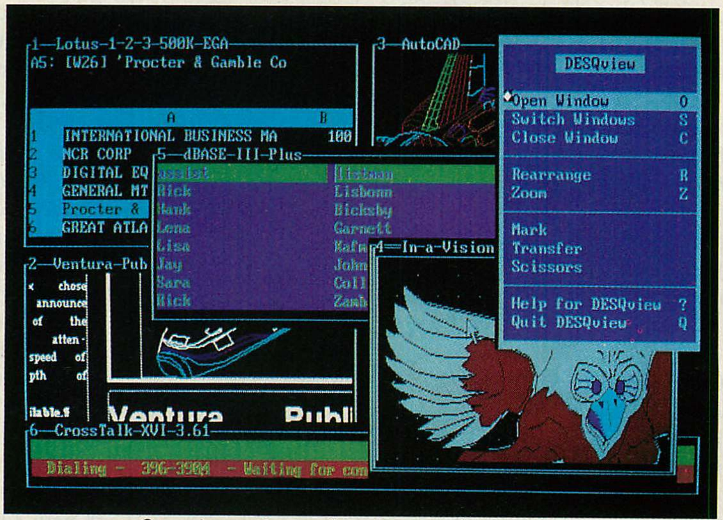
SRMS supports branches; however, the operator must explicitly state his intent to create a branch when the get is performed. If he performs a checkout for edit on any nontip revision without specifying that a branch is to be created, SRMS assumes that he wants to use that revision as a base to create the next revision at the tip of the trunk. It would seem that this runs contrary to the idea that deltas are applied one after the other without skipping intervening revisions. Getting a non-tip revision for edit should imply the desire to create a branch.

Files that are retrieved without edit permission are not set as read-only. Also, a file can be checked in even if no revision of that file is out for edit. This and the lack of a user ID limit the usefulness of SRMS on a project with several programmers. SRMS allows only one revision of a file to be out for edit at a time. With SRMS, a distinction is made between taking a file out for editing and locking it. The user must explicitly specify the request for a lock. If he does not, another programmer can take out a revision for edit and will not know that module is already out. This second get operation could change the contents of the **.px** file. Unless each programmer uses a separate directory, unexpected results could occur when the first programmer performs a delta of modifications. Using separate directories, the first programmer would receive an error message when attempting to delta the modifications. An SCMS that supports revision locking should automatically generate the lock at the time that a file is taken out for edit. No convenient method is available to remove a lock on a file. Removing the lock requires the user to manually reset the read-only attribute on the associated **.px** file, then delete the lock file.

Often it is necessary to create a new major release by checking out a file for edit and returning it unchanged at the new major release level. SRMS does not provide a way to force the check in of an unchanged file. The message generated when this is at-

How to tell the difference between DESQview™ 2.0 and any other environment.

Selecting DESQview, the environment of choice, can give you the productivity and power you crave, without the loss of your old programs and hardware. If you like your existing programs, want to use them together, transfer data between them, print, sort, communicate with or process-in-background, yet still have the need to keep in place your favorite PC(8088, 8086, 80286 or 80386), DESQview is the "proven true" multitasking, multi-windowing environment for you. Best of all, DESQview 2.0 is here now, with all the money saving, time saving, and productivity features that others can only promise for the all-too-distant future.



One picture is worth a thousand promises.

And with DESQview's new graphics enhancements for Hercules, CGA, EGA, and VGA, Version 2.0 still offers the same award winning and pioneering features for programs that earned DESQview its leadership, only now you can also run desktop publishing programs, CAD programs, even GEM-™ Topview-™ and Microsoft Windows-™ specific programs. In some cases you'll add as little as 10–40K to your system overhead. Now you can have multi-tasking, multi-windowing, break the 640K habit too and still get an auto dialer, macros, menus for DOS and, for advanced users, a new complete application programmer's interface capability. No wonder that over the years, and especially in recent months, DESQview, and now DESQview 2.0 have earned extravagant praise from some of the most respected magazines in the industry.

INFO WORLD
REPORT CARD
DESQview 2.0
9.1
July 6, 1987

"Product of the Year" by readers vote in InfoWorld.

"Best PC Environment" by popular vote at Comdex Fall in PC Tech Journal's "System Builder" Contest.

"—I wouldn't want to run an IBM or compatible computer without DESQview"—InfoWorld, Michael Miller.

"A colossus among windowing environments"... "will run almost anything"—PC Week, Marvin Bryan.

"Windows, promises, but DESQview delivers"—MICROTIMES, Birell Walsh.

No other environment has consistently pioneered features, openness, and productivity. See for yourself. Send in the coupon. The possibilities are endless with DESQview 2.0.

Attention Programmers: For more information about Quarterdeck's API, and future 386 program extensions, call us today.

SYSTEM REQUIREMENTS
IBM Personal Computer and 100% compatibles (with 8086, 8088, 80286 or 80386 processors) with monochrome or color display; IBM Personal System/2 • Memory: 640K recommended; for DESQview itself 0-145K • Expanded Memory (Optional): expanded memory boards compatible with the Intel AboveBoard; enhanced expanded memory boards compatible with the AST RAMpage • Disk: Two diskette drives or one diskette drive and a hard disk • Graphics Card (Optional): Hercules, IBM Color/Graphics (CGA), IBM Enhanced Graphics (EGA), IBM Personal System/2 Advanced Graphics (VGA) • Mouse (Optional): Mouse Systems, Microsoft and compatibles • Modem for Auto-Dialer (Optional): Hayes or Compatible • Operating System: PC-DOS 2.0-3.3; MS-DOS2.0-3.2 • Software: Most PC-DOS and MS-DOS application programs; programs specific to TopView 1.1, GEM 1.1 and Microsoft Windows 1.03 • Media: DESQview 2.0 is available on either 5¼″ or 3½″ floppy diskettes

Rush me DESQview 2.0! Today! PCT-9

No. of Copies	Media 3½″/5¼″	Product	Retail Price ea.	Total
		DESQview 2.0	$129.95	$
		Shipping & Handling USA / Outside USA	$ 5.00 / $ 10.00	$
		Sales Tax (CA residents)	6.5%	$
			Amount Enclosed	$

Payment: ☐ Visa ☐ MC ☐ AMEX ☐ Check

Credit Card: Valid Since ____/____ Expiration ____/____

Card Number: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Credit Card Name ____

Shipping Address ____

City ____ State ____ Zip ____ Telephone ____

Mail to: Quarterdeck Office Systems, 150 Pico Boulevard, Santa Monica, CA 90405.
NOTE: If you own DESQview call us for a special upgrade offer, or send in your DESQview registration card. AST Special Edition users included.

DESQview™
Quarterdeck

Quarterdeck Office Systems • 150 Pico Boulevard, Santa Monica, CA 90405 • (213) 392-9851

DESQview is a trademark of Quarterdeck Office Systems. AboveBoard is a trademark of Intel Corporation. Hayes is a trademark of Hayes MicroComputer Products Inc. IBM, PC, Personal System/2 and TopView are trademarks of International Business Machines Corporation. Microsoft Windows and MS are registered trademarks of Microsoft Corporation. Mouse Systems is a trademark of Metagraphics/Mouse Systems. RAMpage is a trademark of AST Research, Inc. GEM is a trademark of Digital Research. Hercules is a trademark of Hercules.

CIRCLE NO. 137 ON READER SERVICE CARD

tempted says only that the operation was not performed.

SRMS is another system that does not preserve DOS EOF marks when a file is retrieved from the library. In addition, an empty line just prior to the Ctrl-Z in the original file was deleted when the file was retrieved with a get.

Symbolic version labels are supported by the **sclass** program, which allows new labels to be defined and modules to be added or deleted from the association. An explicit revision number can be given when adding a module or it can default to the latest revision. Unfortunately, it is not possible to add several modules to a class with one **sclass** command. Each module is added separately. Class information is kept in a **classinfo.srm** file.

It is also possible to delete an entire class association. The prompt generated is a little unnerving, though, because it asks if the operator if he wants to "delete class and ALL associated modules." The manual specifies that the actual modules will not be deleted, but most users will hesitate before striking the Y key to continue. A **cget** program retrieves all modules in a specified class. This is tremendously helpful when rebuilding a prior version of a system. **Sclass** also can display a list of all associations in effect for a given symbolic version.

SRMS includes the key-word substitution feature with several key words for file ID, revision ID, and time/date stamps. The default for a get operation is no key-word expansion. This is reasonable because when expanded, key words are totally substituted that would require editing to put them back if the substitution were performed with every get operation. The revision description log also can be included by get. This is accomplished by a command line option, and the revision description can be placed only at the end of the source file. SRMS inserts the revision description using comment symbols appropriate for the language in use based on the extension of the archive file name. A default symbol is used if the extension is not recognized.

Several auxiliary programs are provided for miscellaneous source code management operations. **Prthis** will print a revision history for one or several modules. This listing shows time/date stamps for the revision, delta information, and the revision comments. The user has no control over formatting the output or contents other than to specify module names and a date qualifier. **Srmsinfo** generates a listing of the state of the SRMS library with respect to files that are out for editing or that are locked. This program does not accept a module name as a qualifier. When **srmsinfo module.c** was accidentally run to display information about this single module, the program generated the status report using **module.c** as an output file. This destroyed the working copy of this source file in the process. All modifications performed since it was checked out were lost.

Sdiff produces a listing of differences between two revisions of a file, a revision in the library and a disk file, or two disk files. No utility is available to take this difference listing and generate a new file starting from the base file. **Smerge** can be used to merge two revisions of a file into one single file.

Out of this competent field, TLIB and PVCS are set apart by their ease of use, abundance of features, and ability to be configured.

Performance test results for SRMS make it to be average in speed and size of the archive. It was the slowest system to generate the delta from the 50KB file in test 3. At least it has a continual output to the screen so the operator can track progress.

Seidl Computer Engineering. The Seidl Version Manager (SVM) has the most rigid requirements for directory structure of all of the packages reviewed. Source code is maintained in a project directory. All SVM operations must be performed from this directory. SVM creates three subdirectories: ARCH, BASE, and WORK. The ARCH directory contains all of the SVM library archive files. The BASE directory holds a copy of the base revision that is used to generate the delta when a file is checked in. The copies of working source files are contained in the WORK directory. An optional BACK directory can be created to hold backup copies of the SVM archive files, as a safety measure. Most activity, such as editing and compiling, is performed in the WORK directory. To do a get or put operation in WORK requires going to the parent directory level in order to perform the SVM operations. Such required directory switching is inconvenient.

All SVM operations can be performed by executing the individual programs or by running them from the SVM menu shell. Individual operations are selected from the main menu, which causes a submenu for each operation to appear. Additional menus are available for entering the required information for the current operation. Some of these present too much information. An inexperienced user may balk at the description for specifying revision numbers in Backus-Naur notation, or when, in performing a DOS command, he has to choose whether it is a system command or a program command. The text on the menu describes how system commands will load a second copy of DOS, whereas program commands simply will run quicker because they do not.

When a file name is needed, a list of available choices is obtained by entering *.*. For nonfile name entries, a ? is used. During testing, the wrong one was chosen, and the result was the creation of a symbolic version label of ?.

SVM documentation is confusing when it speaks of a file that has been modified and not checked in. SVM refers to this as a file that is "not versioned." This terminology can be confused with symbolic version labels, which SVM also supports.

This system has no user ID mechanism, nor can a lock be placed on any file. When a file is retrieved using **dget**, copies are placed in both the WORK and BASE directories. The version in the BASE directory is used when **dgen** performs the delta operation. It is from the revision of the file in the BASE directory that the delta is generated. Keeping a copy of revisions in the BASE directory speeds up subsequent SVM operations. Both **dget** and **dgen** have options to effect the deposition of files in the BASE directory. Unfortunately, no examples are given showing the use of these options. Also, to back up a system, it is not clear exactly which files must be saved. It appears that saving the files in the ARCH directory is sufficient. Library archive files are maintained with their read-only attribute set. Both **dgen** and **dget** allow multiple files to be processed with one command. A file retrieved with **dget** receives its original time/date stamp, however, the DOS EOF mark is not preserved by the system. Branches can be created by applying a delta to a modified file for which the base revision is not the tip of the development trunk. No program is available for merging a branch back into the trunk.

When a file is checked in, **dgen** prompts for a comment (which can have a maximum 2,000 characters); however, the fine line editing capabilities available when keying in entries on one of the menu screens are not available when the user enters comment lines. After the comment is completed, the user can elect to edit the comment with a user-specified editor. A user invokes this feature by defining an environment variable, EDITOR, to specify the editor to run.

This is the only system that has an option to store deltas with compression. The documentation claims a 30- to 70-percent storage reduction. Test 3 was run in both normal and compressed mode. The archive file in compressed mode was 34-percent smaller. However, a time penalty is incurred when running **dget** and **dgen**, which means almost doubling of the time to perform the delta during test 3. The compression algorithm uses Huffman encoding with tables selected for specific languages based on file extension.

SVM allows up to 100 threads to be defined. This is the name used to refer to a symbolic version label. Individual modules can be assigned and unassigned to these threads using **dgen**. Symbolic version names that can be associated with the threads are stored in the VERSION.LOG file. This is a read-only ASCII text file that is located in the project directory, that is, the parent of ARCH, BASE, and WORK. This is generally the only file in that directory. Documentation is provided on the format of this file, which can be edited if the read-only attribute is cleared. Symbolic version names can be used in place of revision numbers to specify files for **dget**. SVM also comes with a program called **dmake**, which is used for analyzing versions and extracting source revisions from SVM. **Dmake** communicates information to Seidl's SMK automatic program builder about what must be done to build the program.

The report generation capability in SVM is the best of any of the products reviewed. The **dlist** program gives full control over the items of information that will be displayed. The user also has control over pagination of the output and a footer can be specified to be printed with the report. If spaces are in the footer, the entire option, including the **-FMT =** introducer, must be enclosed in quotes, such as "**-FMT = This is the footer**". This is rather unusual syntax for a command line option. This footer can, alternatively, be declared in the environment variable DLISTFTR.

SVM's delta detection algorithm exhibited the same problem with some files that was seen with version 1.3 of PVCS. A large increase in the size of an archive file can result from small changes in a file that has only one unique line. No problems were found with any other files tested. SVM has the fastest times for tests 1 and 2, and it executed test 3 very quickly as well. The close correspondence of this time with the execution time for PVCS in test 3 along with the similar archive size anomaly indicates that the systems may use similar delta detection algorithms. Test 3 shows a 34-percent reduction in archive size when compaction is used. Also, note that some file compression is achieved in the normal case as well since the archive file after two deltas (46,198 bytes) is smaller than the original file. Surprisingly, the time for operation when compaction is performed is shorter than for the normal case.

RESOURCEFUL

All of these products provide the basic functions necessary to source code management. Some are more cumbersome to use than others, requiring special directory arrangements and special care in file naming, and each has its shortcomings. Out of this competent field, the two best packages are Burton Systems' TLIB and Polytron's PVCS, so designated because of their ease of use, abundance of features, and ability to be configured by the user.

TLIB is an excellent SCMS overall. Its archive file sizes are small and the speed of operation is very good. User configuration control gives the capability to adjust most features of the system for particular project needs. It is only lacking in some features, such as multiline editable revision descriptions and support for branches or a full set of key words. Without question, if price is a strong determining factor in selection of a system, TLIB cannot be beat.

In terms of features, PVCS provides everything necessary to a large multiprogrammer project—more than any other package reviewed. No restrictions are placed in the development environment and all aspects of operation can be customized for specific project needs. Power has a price, however, as PVCS is the most expensive purchase. However, its performance was also above average in all three sets.

No matter which package is selected, its usefulness is only as good as the discipline a single programmer or an entire programming staff is willing to exert in continuous source code maintenance. Often, team members balk at the level of control imposed on them and try to finds ways to bypass the system. This might help solve a near-term problem but can create serious headaches in the future. If team members can see the advantages of an SCMS—that it increases productivity and is easy to use—they will be more inclined to use it. Certainly, any one of the reviewed systems is more productive than manually maintaining changes to source code.

Burton Systems Software
P.O. Box 4156
Cary, NC 27519-4156
TLIB 3.0
919/469-3068
CIRCLE 347 ON READER SERVICE CARD

Datalight, Inc.
17505 68th Avenue NE
Suite 304
Bothel, WA 98011
BRICKS 1.23
800/221-6630; 206/486-5226
CIRCLE 348 ON READER SERVICE CARD

H & S Associates
1335 Hotel Circle S
Suite 206
San Diego, CA 92108
Source Code Interactive Librarian (SCIL) 1.24
619/692-9464
CIRCLE 349 ON READER SERVICE CARD

Polytron Corporation
1815 N.W. 169th Place
Suite 2110
Beaverton, OR 97006
Polytron Version Control System (PVCS) 3.0
800/547-4000; 503/645-1150
CIRCLE 350 ON READER SERVICE CARD

Quilt Computing
7048 Stratford Road
Woodbury, MN 55125
Software Revision Management System (SRMS) 3.0
612/739-4650
CIRCLE 351 ON READER SERVICE CARD

Seidl Computer Engineering
3106 Hilltop Drive
Ann Arbor, MI 48103
Seidl Version Manager (SVM) 1.05
313/662-8086
CIRCLE 352 ON READER SERVICE CARD

Jim Vallino is a PC programmer with more than a decade of experience in the areas of microcoding, high-level applications, and assembly language programming.

Recent PC announcements have left Compaq in an enviable position.

Compaq still

In the midst of the clamor surrounding the new IBM® PS/2 personal computers, one thing is perfectly clear to people who really know PC's. COMPAQ® personal computers still work better. They enhance your productivity *within* the industry standard, and give you maximum performance from the world's largest library of business software.

Demand for the 12-MHz COMPAQ DESKPRO 286 has nearly doubled since the PS/2 introduction.

The 12-MHz COMPAQ PORTABLE III is the smallest, most powerful full-function portable there is.

Still the performance leader

COMPAQ personal computers prove superior in overall performance.

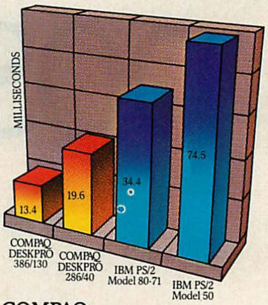
Take speed. The COMPAQ DESKPRO 286® runs your software up to 20% faster than its PS/2™ counterpart. And, the COMPAQ DESKPRO 386™ sets all records for speed in advanced-technology, industry-standard personal computers.

Consider flexibility. Compaq offers 5¼″ diskette drives, and allows you to add 3½″ drives. In fact, you can add up to four different storage devices on all COMPAQ desktops.

Examine compatibility. We let you use all the industry-standard software and expansion boards that you already own.

Look at expandability. Our industry-standard slots enable you to add many extra functions. So you can configure your system exactly the way you want it.

Finally, compare portability. You can't. The 12-MHz 80286-based COMPAQ PORTABLE III™ is the undisputed leader. It offers the performance of a desktop without any of the compromises you'll find in other portables.



COMPAQ fixed disk drives can access data up to 2½ times faster than PS/2 drives.

Compaq applies innovative technology within the industry standard, *without* sacrificing compatibility.

Earn higher returns on your investment

American business has $80 billion invested in

It still simply works better.

works better.

the current PC standard, including 72 million software and hardware products, and hundreds of millions of hours in training.

Compaq designs its computers to protect your investment. And because they do more, they also maximize it.

Two-way compatibility

Compaq has become famous for its legendary compatibility and connectivity. Our personal computers will run thousands of programs, without modification, far faster than other computers. And they work with all the other compatible computers in your office, without time-consuming diskette conversions.

The COMPAQ DESKPRO 386 sets the standard for high-performance, advanced-technology desktop computing.

As for the future, all 80286- and 80386-powered COMPAQ personal computers will run the new MS OS/2 operating system, allowing you to directly access up to 16 megabytes of memory. And they'll run all the applications developed for OS/2™. Again, much faster.

We don't burn bridges, we build them

At Compaq, advances are measured by our ability to push technology forward, without leaving you behind.

The new MS OS/2 operating system will run on all 80286- and 80386-based COMPAQ personal computers.

COMPAQ computers let you incorporate developing technology, and take advantage of the latest technology in a way that's fully compatible with the hardware, software and add-ons you already own. So Compaq protects your investment.

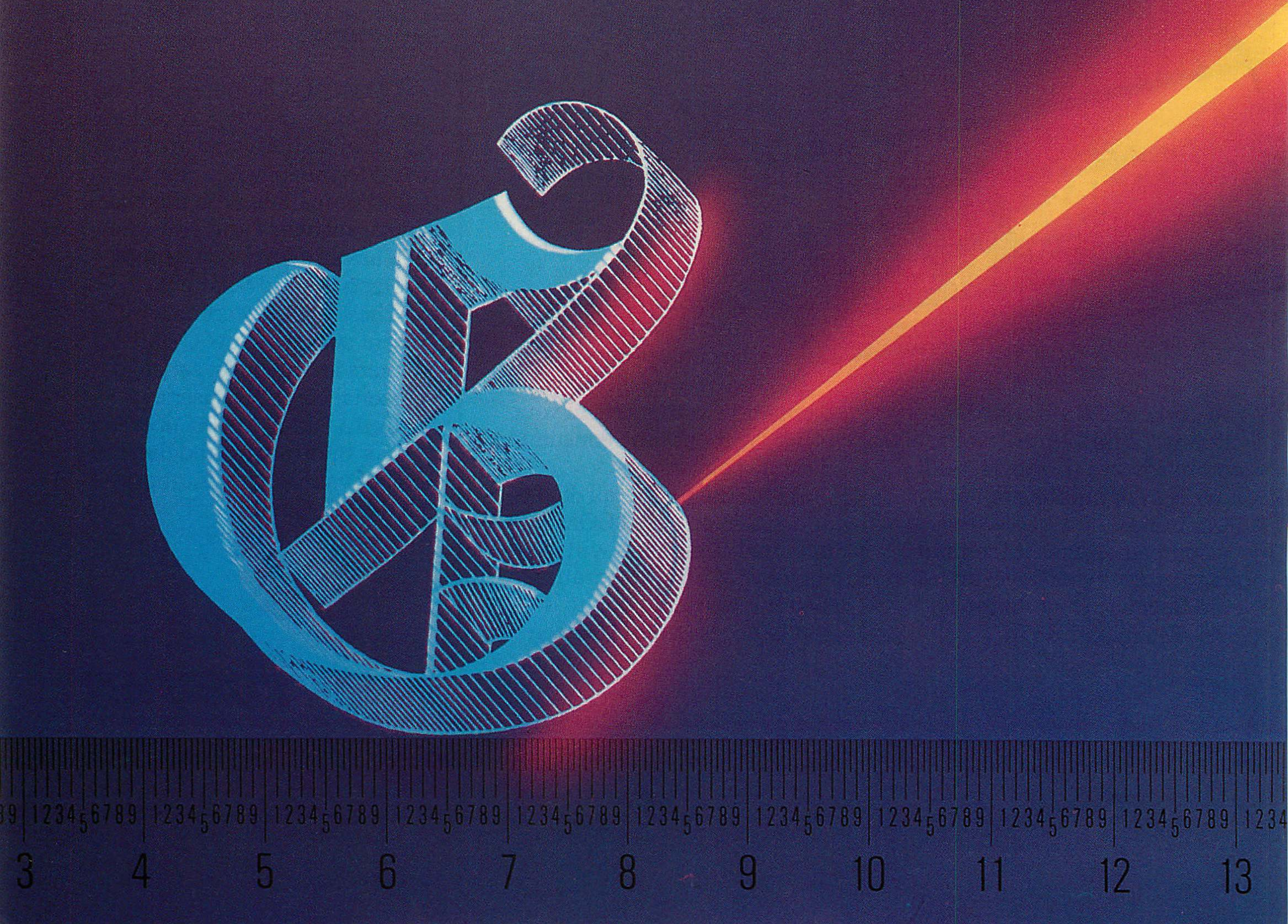
These are all reasons why recent surveys show COMPAQ owners are the most satisfied personal computer users.

Call 1-800-231-0900, operator 39, for information and the location of your nearest Authorized COMPAQ Computer Dealer. In Canada, call 416-449-8741.

COMPAQ®

Laser Metrics

This extensive set of metrics to test printing capabilities determines how laser printers measure up to the HP standard.



RAINER McCOWN and HEETH CLARK

Since their introduction in 1984, laser printers have made the transition from a luxury to a commodity item. The majority of users now install laser printers as high-speed replacements for daisy wheel and other letter-quality printers. In this service they perform admirably, but much of their talent lies untapped. Working in tandem with desktop publishing systems, laser printers are capable of producing printed documents far more sophisticated than those ejected by ordinary letter-quality printers.

Hewlett-Packard was responsible for introducing the laser printer to the PC world; its line is the most prevalent and is considered the standard by which all others are judged. First to come on the scene was the HP LaserJet, followed by the LaserJet+. Recently, HP honed the standard for sophisticated desktop printing applications with its LaserJet Series II. However, it changed very little of the original printer command language (PCL) that the computer uses to drive the printer. HP's PCL has become a standard just as the Hayes AT modem command language is the standard in communications.

Many other manufacturers have introduced laser printers that compete with and try to improve on the original LaserJet models. *PC Tech Journal* has developed a set of software metrics, presented here, to rate the plethora of available printers against the HP standard. A companion article immediately following this one describes how other printers measure up to the HP standard (see "Laser Performance," Rainer McCown and Heeth Clark, p. 100).

PHOTOGRAPHY • LIGHTSTRUCK STUDIOS

THE LASER ENGINE

The laser printer is a byproduct of the personal copier industry. Laser printers are part of a larger class of printers that use a toner transferred to a charged paper or drum. Electrostatic printers, such as the Versatec, are the ancestors of their class, and ion deposition or LED array printers could be considered the next generation.

The laser print engine combines a photoconducting drum illuminated by a semiconductor laser (instead of reflected light) with a copier-style paper handler and toner application and fusing system. Because the photoconducting drum (often called the OPC for organic photoconducting cartridge) has a circumference of less than the length of a sheet of paper, it is cleaned, initialized, and rewritten several times for each sheet of paper passing through it.

The OPC is charged by a corona wire and then discharged by the laser signal. Depending on the type of toner required by the laser engine, the toner adheres either to the area written by the laser (*write-black*) or to the area not written (*write-white*). In the write-black system the OPC is discharged; the laser charges the OPC where black images are required. In the write-white system the OPC is charged to attract the toner over the whole cartridge; the laser then discharges those areas where black is not desired. The paper is charged to attract the toner away from the OPC. In both designs the fuser roller applies heat and pressure to bond the toner to the paper. (The HP LaserJet series are write-black systems.)

The laser printer toner is similar to plain paper copier toners. The formulation of the toner is different with each manufacturer but can be thought of as a fine powder composed of two parts: a colored substance such as carbon for black toner and a wax-like resin that melts when heated, fusing the carbon together, and onto the paper. The toner comes in positive or negative polarity, which describes how it is attracted to the OPC. It also is available in different grain sizes, which limits the ultimate resolution. The smaller grain size of a very fine toner is harder to control and is messy to refill. Only one type of toner is available for any given laser printer.

The various types of toners are optimized for different types of printing. The toner in the Canon cartridge used in the LaserJet series is very fine grained and produces a flat, high-resolution output very similar to offset printing. The toner used by the 6-page-per-minute Ricoh engine has a thicker, almost plastic consistency that gives the output an engraved appearance; the black areas are solid and slightly shiny with excellent coverage of even large areas; however, fine details are less perfect than they would be with a finer-grained toner.

The write-white system produces darker and more consistent blacks whereas the write-black system produces better detail. Detail is increasingly important when printing characters that use very thin lines. A single dot in a write-white system is *reduced* by about 20 percent from the defined size of 1/300 of an inch, while in a write-black system, as used by the HP LaserJets, the size is *increased* by about 20 percent. This variation makes a noticeable difference in gray shading and fine detail of the printed page.

The intended application should drive laser printer selection. Some printers offer better text performance, some better graphics.

Laser printer technology, as with any new technology, uses many new words. Most of these words are borrowed from the typesetting industry. *Font* is the size and shape of the characters in a character set. *Type-style* is a font shape regardless of size. Dot-matrix printers produce simple, readable characters but provide no flexibility in the character shape. Daisy-wheel printers have optional printing elements that add a range of character shapes but little size variation. Laser printers can change both the size and the shape of each character.

Measurements traditionally used by the printing industry have been adopted into laser printer terminology. One inch is divided into 72.27 *points*, and one *pica* is 12 points. Point sizes describe proportionally spaced fonts, and characters per inch describe fixed spacing fonts. Point sizes vary from subscripts of 5 or 6 points to titles of 20 or 30 points. The size for most text is 10 to 12 points which have roughly the same size characters as 12- and 10-pitch characters, respectively.

The terms *portrait* and *landscape* refer to the orientation of print on paper. In portrait mode, text lines are horizontal along the paper's short dimension; in landscape text is read on the long dimension.

PRINTER TRADEOFFS

In selecting a laser printer, the user must consider many variables; some printers may offer better text performance, some better graphics performance. Also to be evaluated are printer memory, interfacing between printer and computer, paper quality and size, maintenance, and multipass printing.

Character quality. One of the primary motivations for purchasing a laser printer is character quality. Figure 1 shows the quality of a character from three different sources. The first sample is from an IBM Proprinter in near-letter-quality mode. Because each dot is printed by a round wire striking the paper, the individual dots remain visible and the paper is dimpled, giving it a slightly coarser appearance.

The HP LaserJet sample shows only slightly higher resolution, but the characters have a smoother, more solid shape than the Proprinter's. After the dots used to form the characters are deposited on the OPC, the entire character shape picks up the toner and is fused to the paper at one time. Resolution varies from printer to printer, primarily due to the type of printing system used (write-white or write-black) and a combination of toner formulations. The high contrast of the black toner sitting on top of the paper also results in laser printouts having an embossed appearance.

By way of comparison, phototypesetters use a much higher resolution with high contrast to produce the definitive image; during the printing process the printer's ink smoothes the edges of the characters.

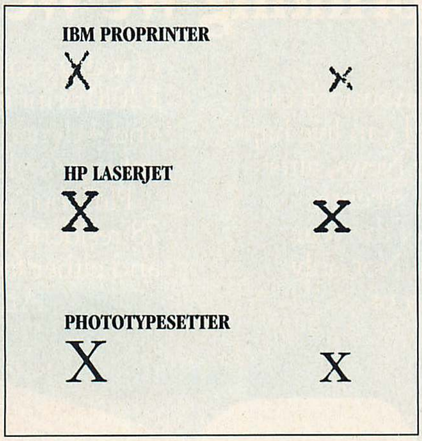
Printer memory. Each text character on a page requires 4.25 bytes of printer memory. A raster graphic requires a single bit of memory for each bit of the graphic. Soft fonts (those that are downloaded from the computer) are stored in the printer's memory where they can be referenced later. During the printing process, after a page has been downloaded and the printer is actually printing, any available spare memory is used as a print spooler allowing the next page to be input while the current page is being printed.

Laser printers are page printers, meaning that the entire page of data must be in the printer's memory before it can be printed. A standard 8.5-by-11-inch page at 300 dots per inch

(dpi) consumes about 1MB of memory. The printer automatically ejects a full printed page, or if the memory on the printer is full before the full page is sent to it, a page is ejected. Printing a sentence, a short list, or a screen requires the operator to force the page to eject. This can be done by pushing a sequence of buttons on the printer, by software commands (for example, form feed or print page), by filling the printer's memory, by sending a carriage return when at the bottom of the page, or by changing the printer's modes (such as switching from portrait to landscape mode).

The original HP LaserJet has limited raster graphic capability due to insufficient memory. It does not have built-in vector graphics functions to draw lines, boxes, and circles, although some other early laser printers do. The HP LaserJet Series II offers an optional memory expansion module with enough memory for a full page of pixel data at the highest resolution of 300 dpi. Most current desktop publishing packages need 512KB of memory. Any printer that does not have this minimum amount is not likely to be fully functional with its emulation.

FIGURE 1: *Character Quality*



The dot-matrix output from the IBM Proprinter is improved on by the HP laser printer, but the quality does not compare with that of the same character produced by a phototypesetter.

Interfacing. The three different methods of interfacing a printer to a PC are serial, parallel, and direct memory access (DMA). The original LaserJet printers had a serial interface to be more compatible with the older daisy-wheel printers, but a parallel interface was available as an option. The Series II has both interfaces as standard. A DMA printer must be supplied with special driver software before the standard PC operating system can drive it.

DOS assumes that printers use a parallel interface. The DOS provisions for serial printers are not very effective and can be used only if the software allows DOS to control the printer (most word processors and all print spoolers do not). A parallel interface is capable of speeds of 100KB per second, although the actual rate is usually limited by software and is quite a bit slower than the maximum hardware rate. It is nonetheless usually considerably faster than using a serial interface.

The IBM parallel printer interface is a superset of the Centronics protocol, which provides for only a few status signals to be returned to the computer. This lack of status signals is made worse because so many of the compatible printer boards do not emulate the full IBM printer interface standard but only the Centronics subset. This lack of information reduces the corrective action that a program on the PC can take to control the printer.

A serial interface is bidirectional. Status and error messages can be sent to the PC from the printer in the same way as in the Apple LaserWriter and other Postscript printers. The LaserJet printers do not send any data back to the PC except for XON/XOFF flow control. Using a serial interface provides greater flexibility in printer location; a printer can even be driven remotely via modem (after the connection is first established using a regular communications package). Cable lengths of 50 feet are specified but can be stretched with inexpensive line drivers.

A serial interface is much slower than a parallel interface. Using a serial interface connected at 9,600 bits per second (bps) requires more than 18 minutes just to transfer the data to the printer. A page of text, usually less than 5,000 characters, takes 5 seconds to transfer to a serial text printer.

The DMA interface, such as used by the Tall Tree's JLaser card, is the fastest interface. Adding a JLaser card can turn the earlier LaserJets (or any other Canon LBP-CX engine-based printer) into a DMA printer; the printer electronics and memory are moved into the PC, leaving the printer to serve only as the mechanical print engine. This approach uses the PC's processor to set up a page to be printed, often dipping into expanded memory to store the page of printout. The data are sent to the printer at the actual print rate (.94 Mbits per second).

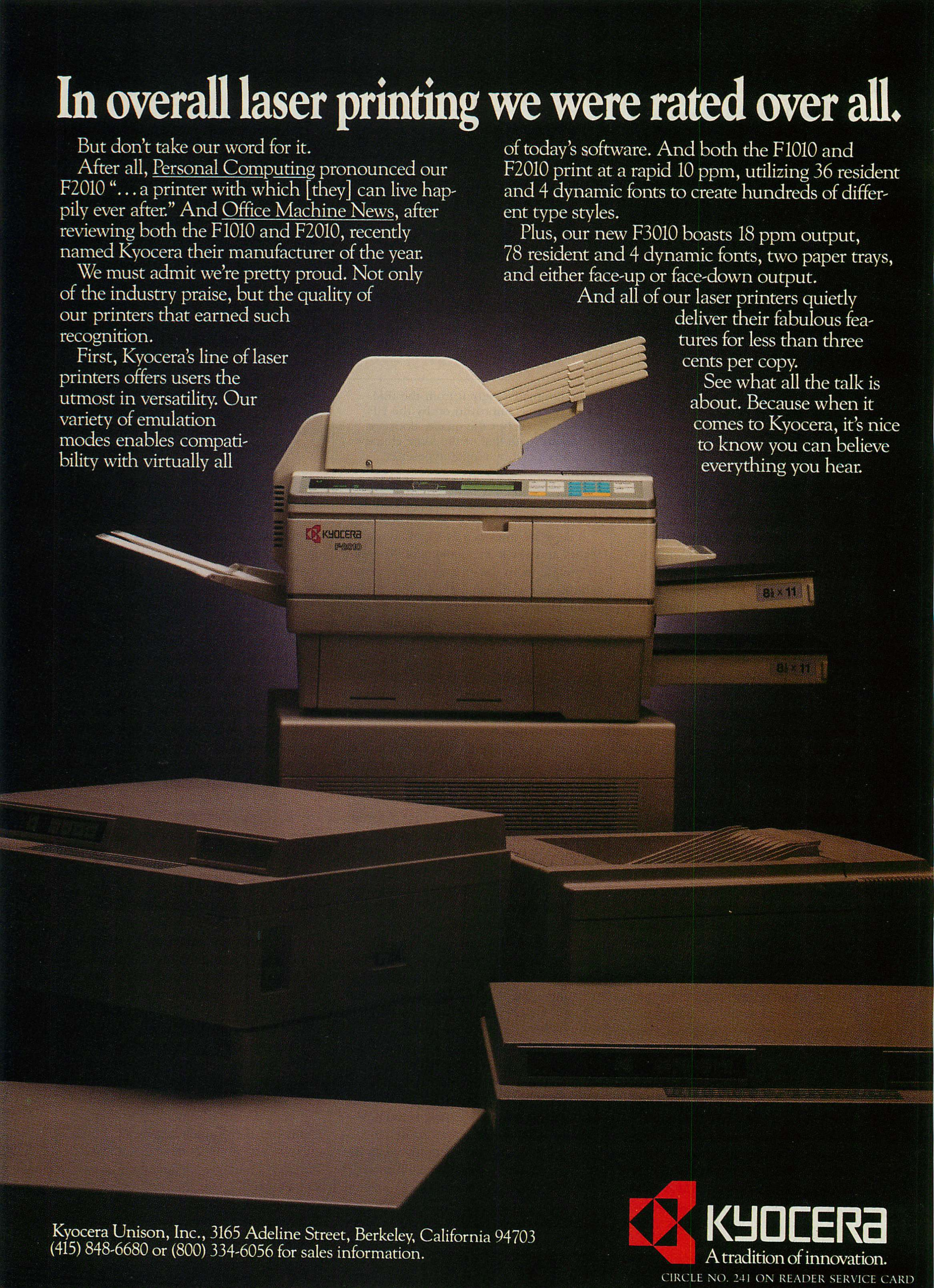
The disadvantages of this approach are several. Moving the printer to another machine is much more complicated than moving a cable. Modem connections are impossible, because there is no serial interface. The PC processor is dedicated during the printing cycle and special drivers must be installed to make the DMA interface appear like a normal printer to the software. Some software packages, such as Xerox Ventura Publisher and Z-Soft Publisher's Paintbrush, supply drivers for the JLaser's DMA interface.

Paper quality. Copier technology was adapted for the early personal laser printers; thus the paper handling limitations of copiers were transferred to laser printers. The mechanical paper handling, the toner adherence, and the heat fusion process place restrictions on the media used. Copier paper (20-pound bond) works well in laser printers. Slightly thinner (15-18 pound) to slightly heavier (30-35 pound) paper also can be used. More expensive paper, such as that used for letterhead, with 25-percent cotton and slight texturing, also works well. Papers specially designed for laser printers have a clay, varnish, or plastic coating to give excellent, eye-pleasing results.

Labels, viewgraphs, envelopes, and heavy paper require a straight-through paper path. Both the LaserJet and LaserJet+ printers provide this path using a single sheet input slot in the rear of the printer with the printed paper ejecting out the front. On the Series II printer, however, this is reversed. The Series II has an optional straight-through output slot in the rear, and the paper is fed in from the front.

In the original LaserJet, the paper emerges with the print side up, so a document is produced with the first sheet at the bottom of the pile. A glance at the paper output pile reveals the progress of the printout and quickly shows any problems such as printing too light, too dark, or running out of toner. The Series II printer outputs the paper upside down so the final document when removed (and turned over) is in the correct order, but gone is the convenience of visual access during the printing process.

Paper size. The current crop of laser printers is designed to print on pages that are 8.5 inches wide. Because the OPC drum circumference is less than the page length, the printer uses multiple drum revolutions during one print cycle. Thus, there is no mechanical limit to the length of the paper that can be printed, but the internal memory and electronics of these printers limit the paper length to 14 inches. If


In overall laser printing we were rated over all.
But don't take our word for it.
After all, Personal Computing pronounced our F2010 "...a printer with which [they] can live happily ever after." And Office Machine News, after reviewing both the F1010 and F2010, recently named Kyocera their manufacturer of the year.
We must admit we're pretty proud. Not only of the industry praise, but the quality of our printers that earned such recognition.
First, Kyocera's line of laser printers offers users the utmost in versatility. Our variety of emulation modes enables compatibility with virtually all of today's software. And both the F1010 and F2010 print at a rapid 10 ppm, utilizing 36 resident and 4 dynamic fonts to create hundreds of different type styles.
Plus, our new F3010 boasts 18 ppm output, 78 resident and 4 dynamic fonts, two paper trays, and either face-up or face-down output.
And all of our laser printers quietly deliver their fabulous features for less than three cents per copy.
See what all the talk is about. Because when it comes to Kyocera, it's nice to know you can believe everything you hear.
Kyocera Unison, Inc., 3165 Adeline Street, Berkeley, California 94703
(415) 848-6680 or (800) 334-6056 for sales information.
KYOCERA
A tradition of innovation.
CIRCLE NO. 241 ON READER SERVICE CARD

continuous forms are tried, the printer assumes the paper has jammed.

Printers are designed to leave a margin on all four sides of the paper. The Canon 300-dpi engine (used in the HP printers) has a nonprinting margin of 48 dots on the left side of the paper, 95 dots on the right side of the paper, and 60 dots each at the top and bottom. On 8.5-by-11-inch paper, the effective print area is 2,407 dots by 3,180 dots (4,080 on 14-inch paper). Thus, the total number of dots displayable is 7,569,182, or 946,147 bytes of data (1,227,570 bytes on a 14-inch page).

Maintenance. Just as dot-matrix printers have ribbons that must be changed, laser printers also have "expendables." The Canon EP cartridge used by the LaserJets and other Canon-engine laser printers is the most popular mechanism for handling the toner and OPC drum. The cartridge is a one-piece unit that prevents the toner from spilling. Many other printers separate the toner from the OPC drum; the toner is contained in a bottle and must be poured in, increasing the risk of spillage.

Separating the toner and OPC results in very little cost savings. The Canon cartridge lists for $115 and is good for about 3,000 copies. Some companies refill a cartridge for about half the price of a new one, or it can be refilled in-house for $20 or less. This refilling is possible because the OPC drum lasts two to three times longer than the toner. The new LaserJet Series II cartridge increases the toner supply and decreases the OPC drum size to result in a matching life span.

The Canon cartridge system has additional advantages. Before moving the printer, the Canon cartridge can be removed, whereas the loose toner systems can create a mess. Also, cartridges have clear advantages for occasional color output. Canon offers a brown and a black toner cartridge. Apple offers several color cartridges that can be used in the LaserJet. Some of the refilling services offer color refills.

Multipass printing. The multipass capability allows for double-sided and multicolor printing. This capability can be used for merging text with graphics or for filling in preprinted forms. Print registration on the page is important for multipass printing, but most PC laser printer paper-feed mechanisms commonly allow variations of 10-pixel diameters, which renders multipass printing somewhat less effective.

Furthermore, the first pass through the printer changes the physical characteristics of the paper, and subsequent passes tend to wrinkle or jam the paper. The fuser roller dries out the paper as it heats it making it less pliable when it passes through the rollers again. Letting the paper relax and absorb water from the air after the first printing greatly improves multipass printing. The relaxation period varies greatly with the temperature, the humidity, and the particular printer.

LASERJET COMPATIBILITY

The *PC Tech Journal* LaserJet Software Metrics perform three functions: they identify the hardware differences, including resolution, speed, page size, and coverage (the amount of the page that can be printed on), between the LaserJet standard and other compatibles; they describe the capabilities of the LaserJet printers; and they verify the software compatibility of all LaserJet-compatible printers. The subroutines are listed in MISS.C (listing 1).

Personal publishing software only hints at laser printer capability. It does not produce professional typeset quality output.

Each of the programs in the LaserJet Metrics series tests at least one basic function. Other functions are also exercised to demonstrate interactions between functions. At minimum, the compatible printer should meet the HP standard for a specific function. The manner in which a printer exceeds the standard should be considered at two levels. If better performance results, such as the ability to print more fonts on a page, then exceeding the standard is worthwhile. If only the applications developer benefits, then a dangerous situation could develop. For example, the LaserJets allow two levels of nesting of their macros; some compatible printers may allow more. A developer who writes an application to include more than these two levels cannot assume it will run on the LaserJet, although it may run on an HP-compatible.

LaserJet drivers are included with most of the popular software packages. WordPerfect, Microsoft Word, Lotus 1-2-3, Lotus Freelance, Publisher's Paintbrush, Software Publishing's Harvard Professional Publisher, Aldus PageMaker, and Ventura Publisher were tested against the HP LaserJets. These personal publishing software packages only hint at the capabilities of laser printers; they do not currently produce professional typeset-quality output. Fully capable typesetting packages are just now becoming available for PC operation, as typified by the public domain $\mathrm{T_EX}$ program. A $\mathrm{T_EX}$ compiler was used to generate some of the sample data files for this article.

The LaserJet Metrics test the five sets of command extensions built into the LaserJet printers: macros, push/pop, rules and patterns, raster graphics, and downloadable fonts. Tests also are included for printable page size and page complexity. No test was devised for the HP hardware font cartridges because they are proprietary. To run the metrics, the executable file should redirect the output to the printer port using the DOS redirector (>).

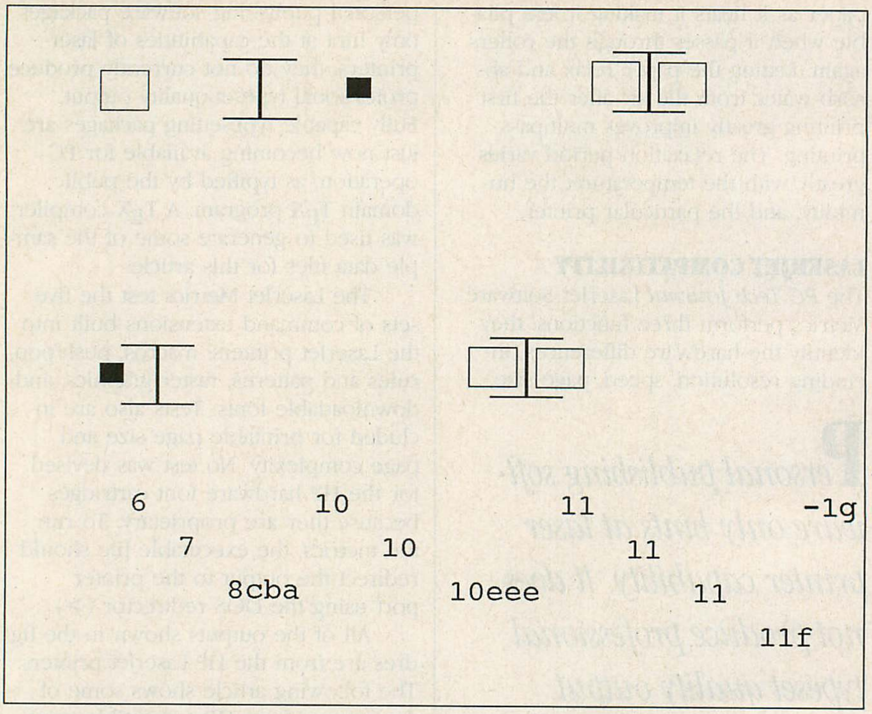
All of the outputs shown in the figures are from the HP LaserJet printers. The following article shows some of the outputs from HP-compatible printers. Due to the nature of printing technology, the laser outputs could not be accurately reproduced in all cases. Subtle differences in resolution and shading may appear more distinct in actual laser output than as reproduced here. For space considerations, the output in the figures may not have exactly the same layout as the actual output produced by the metrics.

Macros. A macro is a single or a group of commands given a single name and executed as a unit. Each macro is assigned an ID number. HP does not specify a range of numbers to be used for this purpose, but during testing the LaserJet accepted a variety of numbers, including −1, 0, and 32768.

The program MACRO.C (listing 2) tests various aspects of using macros. These include calling a macro from within another macro and nesting macros (that is, calling a macro that calls another macro). Because HP states in its Series II documentation that it supports only two levels of nesting, the metric also tests how the printer responds when asked to do three levels of nesting. A recursive test of calling a macro from itself also is included.

The test results for this metric are shown in figure 2. The program first defines three macros: macro number 1 generates an open box, number 2 generates an I-shaped bar, and number 0 generates a solid box. Next the program defines macro number 3, which consists of positioning the cursor and

FIGURE 2: *Macro Output*



The PCL allows the definition of macros. Only two levels of nesting are supported. Requests made to the third level of nesting are ignored by the HP.

calling macro number 1, repositioning the cursor, calling macro number 2, repositioning the cursor and calling macro number 1 again. This results in an image of an I-bar with two open boxes on either side of it. In testing two levels of nesting, macro number 32768 calls macros 0, 2, and 3, resulting in a solid box, followed by the I-bar, followed by the I-bar again with two open boxes on either side.

The program next tests the response of the printer when it is asked to perform more than two levels of nesting. The cursor position is set and macro 6 is executed. This prints the number 6 and calls macro 7, which prints the number 7 and calls macro 8, which prints the number 8 and calls macro number 9. Because the LaserJet supports only two levels of nesting, macro 9 is not called; instead, the macro returns and finishes macro 8 by printing the letter *c*, followed by macro 7 and macro 6, which print the letters *b* and *a*, respectively. Note that when the LaserJet prints a text character it increments the cursor position so that the next character is not printed on top of the first. On a printer that can support additional levels of nesting, macro number 9 would be executed, printing the number 9 and letter *d*.

In order to test the maximum number of levels that can be supported, MACRO.C includes a recursive macro test in which macro number 10 positions the cursor, prints the number 10 and then calls macro number 10 again. On the LaserJet machines this test terminates as expected after three times (two levels of nesting).

MACRO.C performs two more tests. Macro number 11 sends its definition to the printer (print the number 11), redefines itself (move cursor and print the number 11), and then sends the new definition to the printer. The macro is redefined three times before ending by printing the letter *f*. Series II documentation indicates this function is not be supported, but it performed the test as written. The final MACRO.C test simply defined and executed a macro with the ID number −1.

Push/Pop. The program, PUSHPOP.C (listing 3), tests the functioning of the PUSH and POP commands and demonstrates what happens when the maximum number of pushes is exceeded. The PUSH command is used to save the current *X* and *Y* locations on a stack that is 20 values deep. The POP command then recovers each saved location in a last-in, first-out sequence.

PUSHPOP.C first builds a raster graphics box in a macro for use later. It then positions the cursor, saves the current cursor location using the PUSH command, and then executes the macro, drawing a box at the current location. The cursor is then moved to the right and another PUSH command is executed. This pushes the first cursor location down a level and stores the second cursor location on top. Using the current cursor location, the macro is executed drawing another box. This sequence is repeated until 22 boxes appear across the page. A total of 21 PUSH commands have been executed.

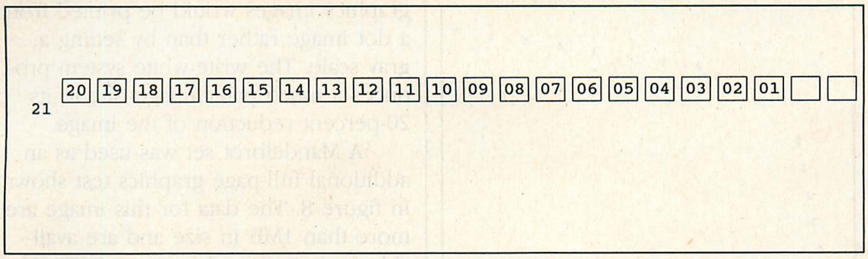
The LaserJet can handle only 20 levels of pushing, however, so to determine the effect of an extra push, the stored locations are popped in turn and an incrementing text number is printed at those locations. As shown in figure 3, the LaserJet ignored the PUSH command after the twentieth level; thus, when the first POP command is executed, the current cursor location is the twentieth square across the page and the number 01 is printed in that square. The next POP command moves the cursor one square to the left and the number 02 is printed. The twentieth POP command gives the location of the leftmost square and the number 20 is printed. The twenty-first POP command has no effect, so the number 21 is printed on the next line.

A printer that is not HP compatible may continue to accept pushes beyond the 20-level limit, causing the numbers to appear in a square to the right of their correct position, or it may replace the last stored cursor location with the extra PUSH command's location causing only the first number to be in an unexpected position.

Complexity. Because of memory limitations, LaserJet printers format the line of dots for the laser in realtime (as it is actually printing the page) from all of the objects (characters, raster graphics, rules, macros, etc.) that are specified for that line of the page. The page that is sent to the printer page is not necessarily a page of dots, but can also consist of rules, macros, or characters that have to be translated by the microprocessor on the printer itself. When the internal microprocessor can no longer keep up, the printer reaches its complexity limit. HP explains that certain parameters, such as macros and vertical lines, are more processing intensive than others. The complexity test program, COMPLEX.C (listing 4), prints a large number of vertical rules on a page; the rules start at various vertical offsets gradually increasing the complexity to beyond the printer's limit.

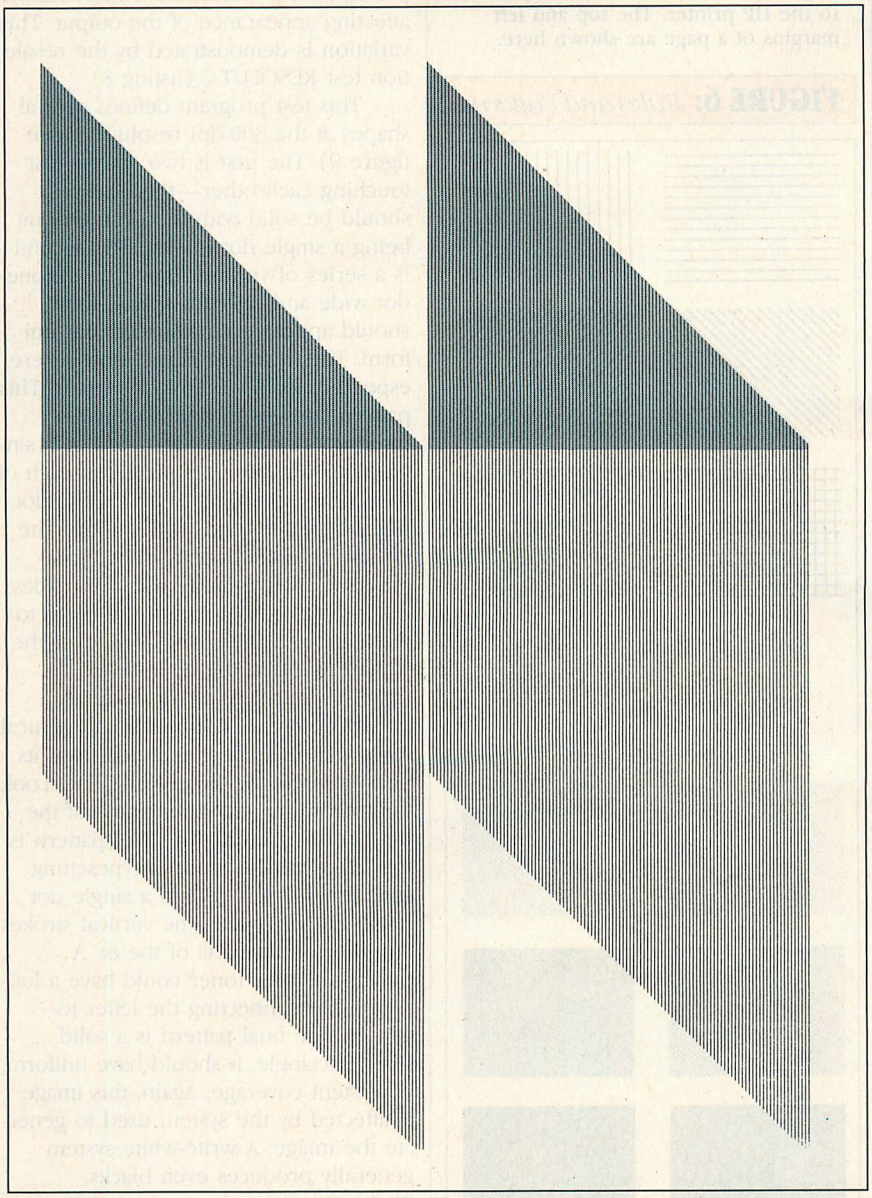
The LaserJet reaches its limit when this test is performed (see figure 4). The printer signals an error message

FIGURE 3: *PUSH/POP Output*



In the PUSH/POP metric, the 22 graphics boxes are specified on the page first. The LaserJet supports only up to 20 pushes and then ignores the twenty-first push. This means that the number 01 prints in the twentieth square, and the number 21 prints on the next line. This metric is found in listing 3.

FIGURE 4: *Complexity Output*



Complexity is one of the areas in which a compatible printer can improve on the HP standard. The HP printers reach their limit after printing only the top portion of the full design created by the complexity metric. The correct output for this metric is two parallelograms as shown here; the HP's attempt is shown in the tinted area. It gives an error message, rejects the page, and must be reset manually.

and ejects the page, necessitating manual resetting of the computer, particularly undesirable in a network where the error message may not be apparent. Reducing the complexity of the page produces the problem at a point where more of the page has printed before the error message occurs.

Complexity is one of the areas in which a compatible printer can improve on the HP standard. However, a page laid out on the compatible machine will not necessarily print on the LaserJet. Because the specification of complexity is vague, no other printer can be totally compatible in this area.

Printable page size. The LaserJet printers leave an unprintable border approximately 0.25 inches wide on all four sides of the paper. These blank zones are designed to reduce the amount of toner that misses the paper. This test program, PAGESIZE.C (listing 5), uses X and Y positioning commands (followed by a character) to print a character at positions on both sides of the limit of the defined page in order to establish the actual printable page limits. The LaserJet printers do not print a character that extends into the border area . The 0, 0 point on the page is the upper left printable point on the page while the 0, 0 point of a character is the lower left corner of the character box.

The program attempts to write characters closer and closer to the edges of the page by changing the cursor position. Negative positions are not allowed; however, negative relative offsets can be used. First the top edge is tested by printing @ at a position with a *Y* coordinate of 230 and then the numbers 1 through 9 and 0 with the *Y* cursor position reduced by using a negative relative offset. Only the numbers 0 through 7 are actually printed; the others are out of range and are ignored. A similar test is performed for the other three sides with a similar result for the bottom and right sides. On the left-hand edge, however, the printer assumes that the requested position is 0 instead of out of range as on the other edges and prints the characters that were ignored on the other edges against the left margin. Figure 5 shows the top and left sides of this output on the HP printer.

Rules and patterns. HP defines six patterns and eight shades of gray, forming a gray-scale pattern. HP also has defined rules (horizontal or vertical lines) that can vary not only in length but also in width. The test program, RULES.TXT (listing 6) defines a rectan-

gular block and prints this block with each of the six predefined patterns. Then eight more blocks are printed, one with each of the eight shades of gray. Then four black blocks are printed: the first the full size, the second with a one-dot-wide *X* dimension, the third with a one-dot-wide *Y* dimension, and the fourth with both dimensions set to a single dot in size.

Figure 6 shows all of the patterns and gray scales, but not the extra black block, the rules, and the dot. The metric prints three additional blocks that are not included in the figure because the print quality of the magazine would not truly represent the shading. These extra blocks show the evenness of a black area. On the HP LaserJet+ these blocks were not an even black. This unevenness was not as apparent on the Series II printer, which has a different cartridge than the earlier models.

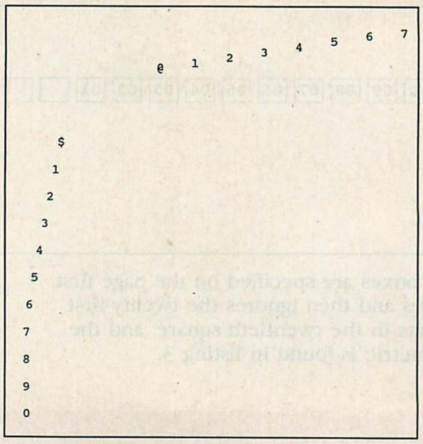
Full-page graphics. The original LaserJet printer was able to do some raster graphics. Raster graphics consume large amounts of memory because each dot in the image area requires one bit of memory. Memory can be saved by using a lower resolution than the 300-dpi maximum. Resolutions of 150 dpi, 100 dpi, and 75 dpi are available. The LaserJet Series II has 512KB of memory standard, and with expansion memory (available in sizes of 1, 2, or 4MB) the Series II can produce a full-page graphic at the 300-dpi resolution.

A full page of graphics is defined as one independent bit, at the printer resolution, for every point on the page. For an 8.5-by-11-inch page, that works out to 8,415,000 bits or 1,051,875 bytes. The printable page size is 8-by-10.5, which means that only 945,000 bytes are required to contain the printable page. If the printer has less memory than is required, the graphic is split by the printer in such a way that the second page begins at the same vertical offset as the first page ended.

The program GRAPHICS.C (listing 7) demonstrates a full page graphic that is truncated at the right and bottom edges. The algorithm used to produce it places an individual dot randomly into each byte sent to the printer. This prevents any data compression by the printer and checks the operation when the printer's memory is exceeded (assuming that less than 1MB available memory). The pattern should appear uniform and printing irregularities evident (see figure 7).

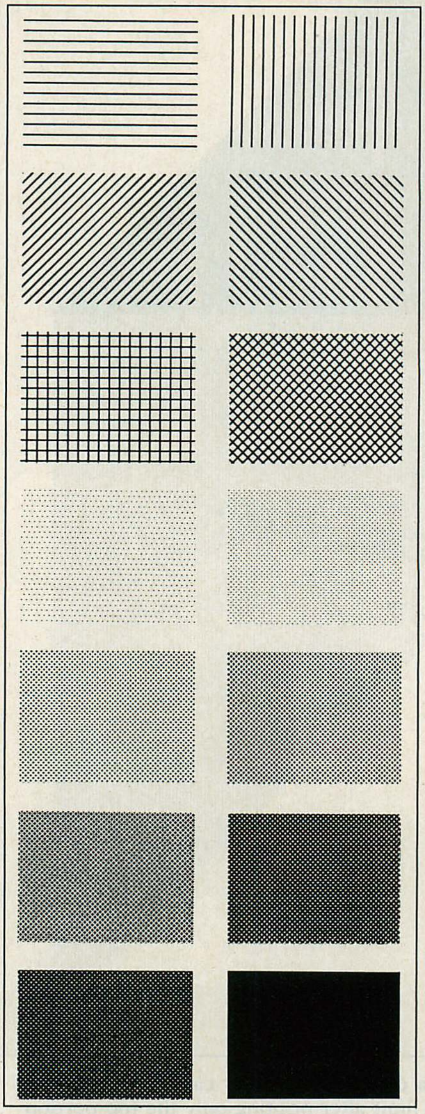
This test helps to illustrate the type of shading that would be seen in a graphics image. The gray scales that are

FIGURE 5: *Page Size Output*



Negative numbers are not acceptable to the HP printer. The top and left margins of a page are shown here.

FIGURE 6: *Rules and Patterns*



Six predefined patterns and eight gray shades are available. These can be used only as rectangular shapes.

defined in the printer can be used only as rectangles, so fill areas in general graphics images would be printed from a dot image rather than by setting a gray scale. The write-white system produces a much paler image due to its 20-percent reduction of the image.

A Mandelbrot set was used as an additional full-page graphics test shown in figure 8. The data for this image are more than 1MB in size and are available in compressed form on PCTECH-line as DATA.ARC.

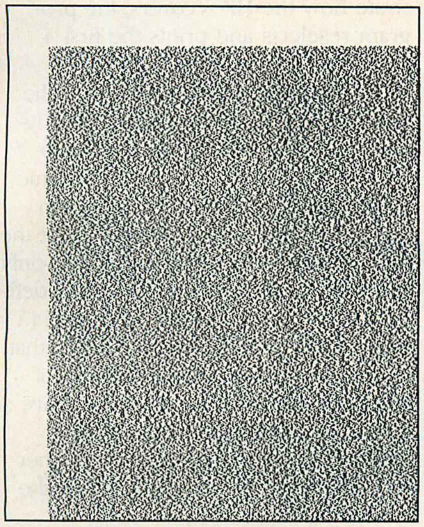
Resolution. Resolution consists of round dots and square dots, dots pushed into the paper and dots sitting on top, and relationships of dots next to each other. The toner used in laser printers varies considerably, noticeably affecting appearance of the output. This variation is demonstrated by the resolution test RESOLUT.C (listing 8).

This test program defines several shapes at the 300-dpi resolution (see figure 9). The first is two arrows just touching each other—the blacks should be solid with the central point being a single dot in size. The second is a series of vertical lines that are one dot wide and one dot apart—they should appear evenly spaced and uniform. The third test is the most severe, especially for a write-black printer. This pattern consists of hollow squares three dots on a side separated by a single dot from similar squares on each of their four sides. The central white dot of each square tends to close up. The HP's write-black system tends to decrease the white area on all four sides, the charge on the central dot tends to dissipate, and the toner flows over the square filling in the open space.

The fourth pattern is individual dots packed into horizontal and vertical rows with each dot separated from its nearest neighbors by a white line. Look for uniformity and distinctness of the dots in this pattern. The fifth pattern is the letter *m* taken from a typesetting style font. This font uses a single dot width neck between the vertical strokes and the curved parts of the *m*. A thicker, heavier toner could have a lot of trouble connecting the letter together. The final pattern is a solid black rectangle; it should have uniform, consistent coverage. Again, this image is affected by the system used to generate the image. A write-white system generally produces even blacks.

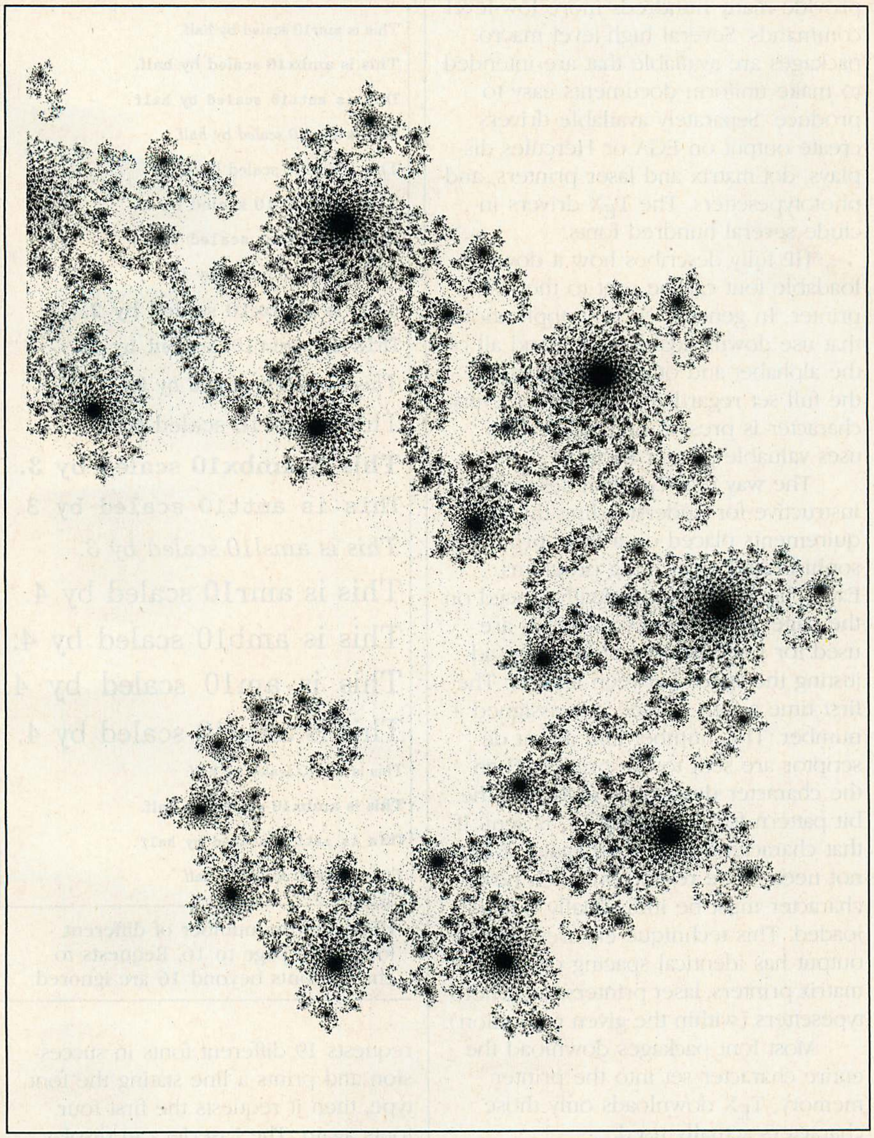
Fonts. Fonts are the most complicated programming feature in the LaserJet printers. The existence of a font downloading feature is important even though it is rudimentary compared, for

FIGURE 7: *Full-page Graphics*



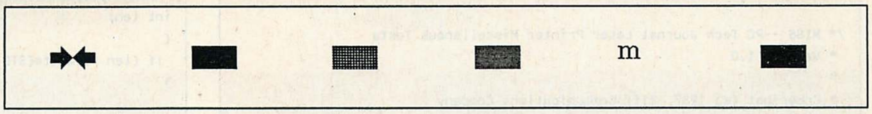
The full-page graphics metric uses random dots to fill the page at 300 dpi. HP's output is uniform; irregularities that occur in other printers will stand out when using this metric.

FIGURE 8: *Mandelbrot Plot*



During printer testing, a Mandelbrot plot was used as an additional test of full-page graphics capabilities. Even though there is no control over the individual dot size, the human eye easily converts dot frequency into intensity levels.

FIGURE 9: *Resolution Output*



The resolution metric produces a series of test patterns at 300 dpi. The toughest test for write-black systems like the HP printers is the third, which consists of hollow squares three dots on a side separated by a single dot from similar squares on each their four sides. The central white dot of each square tends to close up.

example, with the Apple LaserWriter. To download a font to an HP printer, the font must be identified and then each character downloaded individually. The printer is not capable of any font modifications, such as scaling, outlining, or bolding. Rotation must be done externally and the rotated font must be sent again to the printer. Many of the LaserJet font parameters are not completely specified in the HP reference manual, causing compatibility problems for other printers.

The program DOWNLD.C (listing 9) illustrates building a single character for a font and determines if a user-defined font or an internal font is being used. It sets up a new font, downloads the shape of the letter *n* to character number 120 (the ASCII value for the letter *x*), and then prints the letter on the page. If the printer substitutes a different font for the defined font, then the character *x* is printed. If the printer correctly uses the defined font, then an *n* is printed. To have a complete character set in this font, each of the other letters must be individually created and sent to the printer.

Font design requires considerable skill, training, and effort. For example, to change the size of a character set, each character shape must be adjusted to compensate for how the human eye perceives shapes and relationships.

PC TYPESETTING

Sophisticated typesetting programs for the PC are beginning to enter the market, largely spurred on by the creation of T$_E$X, a system of public domain programs developed by Donald E. Knuth at Stanford University, author of "The Art of Computer Programming." He saw the need for a comprehensive typesetting system for manuscripts with mathematical formulas. At least two T$_E$X-adapted compilers are available: PC T$_E$X from Personal T$_E$X, Inc. and MicroT$_E$X from Addison-Wesley.

Unlike desktop publishing programs, T$_E$X is not an interactive system; it is so powerful and exacting that much faster computers would be needed for it to operate in realtime. It

has hundreds of primitive commands plus an extensive macro system that provide many hundreds more low-level commands. Several high-level macro packages are available that are intended to make uniform documents easy to produce. Separately available drivers create output on EGA or Hercules displays, dot-matrix and laser printers, and phototypesetters. The TEX drivers include several hundred fonts.

HP fully describes how a downloadable font can be sent to the laser printer. In general, current applications that use downloadable fonts send all of the alphabet and other characters in the full set regardless of whether every character is present on a page. This uses valuable printer memory.

The way that TEX handles text is instructive for understanding the requirements placed on a laser printer by sophisticated typesetting programs. Each character is individually placed on the page; positioning commands are used for justification and kerning (adjusting the space between letters). The first time a font is used it is assigned a number. This number and a font descriptor are sent to the printer. Then the character descriptor, including the bit pattern for that character, is sent. If that character/font is used again, it does not need to be resent, but each new character must be individually downloaded. This technique ensures that the output has identical spacing on dot-matrix printers, laser printers, and phototypesetters (within the given resolution).

Most font packages download the entire character set into the printer memory; TEX downloads only those characters actually used.

To test the font changing capabilities of laser printers, the data file FONTS.HP (available on PCTECHline) requests 19 different fonts in succession and prints a line stating the font type, then it requests the first four fonts again. The LaserJet can display only 16 fonts on a page (32 fonts can be in memory at one time). Thus, three font changes are ignored by the printer, so the line is printed in the font of the previous line. To demonstrate how the HP recovers, the program reselects and prints the first 4 fonts (see figure 10).

FIGURE 10: *Font Changes*

This is amr10 scaled by half.
This is ambx10 scaled by half.
`This is amtt10 scaled by half.`
This is amsl10 scaled by half.
This is amr10 scaled by 1.
This is ambx10 scaled by 1.
`This is amtt10 scaled by 1.`
This is amsl10 scaled by 1.
This is ambx10 scaled by 2.
`This is amtt10 scaled by 2.`
This is amsl10 scaled by 2.
This is amr10 scaled by 3.
This is ambx10 scaled by 3.
`This is amtt10 scaled by 3.`
This is amsl10 scaled by 3.
This is amr10 scaled by 4.
This is amb10 scaled by 4.
This is am10 scaled by 4.
This is amsl10 scaled by 4.
This is amr10 scaled by half.
This is ambx10 scaled by half.
`This is amtt10 scaled by half.`
This is amsl10 scaled by half.

HP limits the number of different fonts on a page to 16. Requests to change fonts beyond 16 are ignored.

Characters will be missing in the lines where the font changes were ignored. This is because TEX has supplied the printer with only the characters necessary to print the sixteenth line. When the printer tries to print the seventeenth line, it is able to print only as many of the characters as it has definitions for. This problem is due to a page being defined for the printer that has too many font changes. Some of the HP-compatible laser printers have a higher font-per-page limit, but programs that are written to these higher limits will not be compatible with the HP laser printers.

COMPLICATED CRITERIA

At first look, speed, resolution, and HP emulation are obvious considerations in selecting a laser printer. Underneath these considerations, however, lies a more complicated set of criteria: toner type, write-white or write-black, and memory expandability. Each laser printer manufacturer makes tradeoffs in satisfying these criteria, with the intent of attracting business in a price-concious market. Armed with the tools and information presented here, the informed buyer can select the best laser printer for the job.

Rainer McCown is president of Rhintek, Inc., a systems software company that develops products and consults in communications, compilers, editors, and graphics. Heeth Clark is an instructor in computer literacy at The Johns Hopkins University. Bob Smith, president of Qualitas, Inc., contributed to developing the metrics for this article.

LISTING 1: MISS.C

```
/* MISS --PC Tech Journal Laser Printer Miscellanous Tests
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown and Bob Smith
 * Common Routines for C programs.
 */

#include "io.h"
#include "dos.h"
#include "string.h"
#include "fcntl.h"
#define STD_OUT 1

/* SNDL -- Use this routine when the string-to-be-printed
   contains embedded binary zeros (which can confuse
   the STRLEN function used in SND). */

void sndl(sray, len)
char sray[];
int len;
{
 if (len != write(STD_OUT, sray, len)) printf("%s\r\n", sray);
}

/* SND -- Use this routine to send a string to the standard printer */

void snd(sray)
char sray[];
{
 sndl(sray, strlen(sray));
}

/* SETBINARY -- Change a file handle to binary mode to avoid
   converting LFs to CR,LF and to avoid stopping on EOFs */

void setbinary(fh)
int fh;
{
 union REGS inregs;
```

```
/* Change to binary mode via SETMODE
   to avoid converting LF to CR,LF  */

 setmode(fh, O_BINARY);

 /* Change to binary mode via IOCTL
    to avoid stopping on EOF  */

 inregs.x.ax = 0x4400;        /* Fn code to get device information */
 inregs.x.bx = fh;            /* For the file handle */
 intdos(&inregs, &inregs);    /* Return device info in DX */
 if(inregs.x.dx & 0x0080)     /* If it's a device, ... */
   {
    inregs.h.dh = 0;          /* Ensure zero */
    inregs.x.dx |= 0x0020;    /* Turn on binary mode bit */
    inregs.x.ax = 0x4401;     /* Fn code to set device information */
    intdos(&inregs, &inregs); /* Set device info from DX */
   }
}
```

LISTING 2: MACRO.C

```
/* MACRO -- PC Tech Journal Laser Printer Macro Tests
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown
 *
 * This program creates three macros that draw raster graphics.
 * Then other macros are called which in turn call the first
 * macros.  The final test determines the maximum calling depth.
 */

#include "io.h"

#define STD_OUT 1

extern void sndl(char [], int),
            snd (char []),
            setbinary(int);

/******************************** MAIN ********************************/

void main()

{
 int row, cnt, bit;
 unsigned char byte;

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs
  */

 setbinary(STD_OUT);

 /* Initialize the printer */

 snd("\x1BE");               /* Reset the printer   */
 snd("\x1B&l0E");            /* Zero the top margin */
 snd("\x1B&s1C");            /* Disable EOL wrap    */
 snd("\x1B9");               /* Clear margins       */
 snd("\x1B&l0O");            /* Landscape mode      */

 /* Build the macros  */

 snd("\x1B&f1Y\x1B&f0X");    /* Start macro #1 definition */
 snd("\x1B*t150R");          /* Resolution = 150dpi  */
 snd("\x1B*r1A");            /* Start raster graphic  */

 /* Send the bit pattern for a hollow box */

 snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");  /* Top    */
 for (cnt = 0; cnt < 31; cnt ++)       /* Mid    */
      sndl("\x1B*b5W\x80\x00\x00\x00\x01", 10);
snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");  /* Bottom */

/* End the box and the macro */
snd("\x1B*rB\x1B&f1X");

/* Position cursor, execute macro #1 */
snd("\x1B*p100X\x1B*p120Y\x1B&f1y2X");

snd("\x1B&f2Y\x1B&f0X");       /* Start macro #2 definition */
snd("\x1B*t100R");             /* Resolution = 100dpi  */
snd("\x1B*r1A");               /* Start raster graphic  */

/* Send the bit pattern for an I  */

snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");   /* Top    */
for (cnt = 0; cnt < 31; cnt ++)        /* Mid    */
     sndl("\x1B*b5W\x00\x00\x18\x00\x00", 10);
snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");   /* Bottom */

/* End the I and the macro */
snd("\x1B*rB\x1B&f1X");

/* Position cursor, execute macro #2 */
snd("\x1B*p300X\x1B*p100Y\x1B&f2y2X");


/* Start macro #3 definition */
snd("\x1B&f3Y\x1B&f0X");

/* Position and then call macros #1,#2,#1  */
snd("\x1B*p+00X\x1B*p+00Y\x1B&f1y3X");
snd("\x1B*p+35X\x1B*p-82Y\x1B&f2y3X");
snd("\x1B*p+75X\x1B*p-82Y\x1B&f1y3X");

/* End macro #3 definition */
snd("\x1B&f1X");

/* Position cursor, execute macro #3 */
snd("\x1B*p900X\x1B*p100Y\x1B&f3y2X");

snd("\x1B&f0Y\x1B&f0X");       /* Start macro #0 definition */
snd("\x1B*t300R");             /* Resolution = 300dpi  */
snd("\x1B*r1A");               /* Start raster graphic  */

/* Send the bit pattern for a solid box */

snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");   /* Top    */
for (cnt = 0; cnt < 31; cnt++)         /* Mid    */
     snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");
snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");   /* Bottom */

/* End the box and macro #0 */
snd("\x1B*rB\x1B&f1X");

/* Position cursor, execute macro #0 */
snd("\x1B*p500X\x1B*p130Y\x1B&f0y2X");


/* Start macro #32768 definition */
snd("\x1B&f32768Y\x1B&f0X");

/* Position and then call macros #0,#2,#3  */
snd("\x1B*p100X\x1B*p730Y\x1B&f0y3X");
snd("\x1B*p135X\x1B*p700Y\x1B&f2y3X");
snd("\x1B*p700X\x1B*p700Y\x1B&f3y3X");

/* End macro #32768 definition */
snd("\x1B&f1X");

/* Position cursor, execute macro #32768  */
snd("\x1B*p900X\x1B*p100Y\x1B&f32768y2X");

/***************** Start macro depth test ****************/
/* The LaserJet allows a depth of two only */

snd("\x1B&f6Y\x1B&f0X");        /* Start macro #6 definition */
/* Position and then call macro #7  */
snd("\x1B*p+50X\x1B*p+70Y6\x1B&f7y3X");
snd("a\x1B&f1X");               /* End macro #6 definition */

snd("\x1B&f7Y\x1B&f0X");        /* Start macro #7 definition */
/* Position and then call macro #8  */
snd("\x1B*p+50X\x1B*p+70Y7\x1B&f8y3X");
snd("b\x1B&f1X");               /* End macro #7 definition */

snd("\x1B&f8Y\x1B&f0X");        /* Start macro #8 definition */
```

The Leaders Made PVCS
The Leading Source Code Control System.

NOW VAX/VMS & MS-DOS Versions

When it comes to maintaining their most valuable asset, the leading software publishers rely on the POLYTRON Version Control System (PVCS). From accounting firms to airlines, the leading service companies depend on PVCS to maintain the integrity of their programs. Leading manufacturing companies use PVCS to maintain their state-of-the-art software. Leading high technology companies turn to PVCS to handle configuration management for software projects that represent an investment of hundreds of thousands of dollars. The largest aerospace companies and defense contractors use PVCS to maintain integrity of projects during development and after delivery of software. Independent programmers use PVCS to improve their productivity and software quality for themselves and their clients.

Simplify Configuration Management

When large and complex software programs are being developed on personal computers or VAX minicomputers, effective management of the revisions and versions becomes critical. PVCS simplifies this process and lets you effectively control the proliferation of code changes. We used UNIX SCCS and RCS as models. However, our own experience, and the input of hundreds of programmers and managers has enabled us to significantly improve upon these models.

PVCS provides many powerful functions including:

- Storage & Retrieval of multiple revisions of text.
- Maintenance of a complete history of changes.
- Maintenance of separate lines of development using branching.
- Merging simultaneous changes.
- Resolution of Access Conflicts.
- Modules can be retrieved by their own revision number, system version name, or specified date.
- Uses "reverse deltas" to rebuild a prior version making PVCS the fastest version control system over the project life cycle.
- Projects already under development or in the maintenance stage can be easily put under the control of PVCS.

Manages Development On Local Area Networks

Programming teams using Local Area Networks depend on PVCS to help the managers and team members work together. In fact, Novell and 3Com themselves depend on PVCS to manage the versions of their own network software products.

Supports MS-DOS and VAX/VMS Development

Now, companies that develop software on VAX systems running VMS can also use PVCS. And since the VMS and MS-DOS versions of PVCS use the same "logfile" format, you can easily develop software on PCs and maintain the code on the VAX or vice versa. The menu-driven, screen-oriented interface (and optional command-driven interface) makes it easy for programmers and librarians or administrators to use PVCS on a PC or VAX or both systems.

PVCS Maintains System Integrity

PVCS prevents corruption of code that could ordinarily result from security breaks, user carelessness or malfunctions. The levels of security can be tailored to meet the needs of your project.

PVCS & PolyMake Work Together

PolyMake, the leading MS-DOS make utility, is now available for the VMS operating system. This allows you to write makefiles that will function in both PC and VAX environments. Additionally, PolyMake reads time & date stamps of PVCS archives for fast, accurate program rebuilding.

PVCS and PolyMake Maintain Source Code Written In Any Language.

The Preferred Version Control System

The customers listed below are just a few of the innovative leaders that have made PVCS the leading version control program for personal computers.

Alcoa Aluminum
Arthur Anderson
AT&T
Ashton-Tate
Bank of America
Bell Labs
Bendix
Boeing
CIGNA
Citibank
3Com
Colonial Penn
Commerce Clearing House
Control Data Corp.
Corvus
CXI
Digital Equipment Corp.
Deloitte Haskins + Sells
Diebold
Dow
Dunn & Bradstreet
EDS
Educational Testing Service
E-Systems
Equitable Life
Federal Express
First Boston
Ford
Fox Software
Fujitsu
GTE
Hardees
Hewlett-Packard
Honeywell
Hughes Aircraft
IBM
Industrial Networking
Intel
ISC Aerospace
IVAC
Javelin
Lattice
Lawrence Livermore
Lotus
McData Corp.
McDonnell Douglas
Mead Data Central
MIT Lincoln Labs
Nastec
Novell
NCR Technologies
Pitney Bowes
Plexus Computers
Price Waterhouse
ROLM
Rockwell International
Safeco
Sears
Security Pacific
Sperry
Software Publishing
Spacelabs
Standard Oil
Standard & Poors
Tandem
Tektronix
Telex
Texas Instruments
Touche Ross
Unisys
United Airlines
United Parcel Service
United Technologies
U.S. West
Westinghouse Electronics
Xerox

Only PVCS meets the needs of independent programmers and corporations. Once you standardize on PVCS, the archives used to track and monitor changes are interchangeable between any PVCS product. You will receive full credit for your initial purchase if you upgrade to a higher-priced MS-DOS version of PVCS.

Personal PVCS — Offers most of the power and flexibility of Corporate PVCS, but excludes the features necessary for multiple-programmer projects.

Corporate PVCS — Offers additional features to maintain source code of very large and complex projects that may involve multiple programmers. Includes multi-level branching to effectively maintain code when programs evolve on multiple paths (e.g. new versions for different host systems, or a new program based on an existing program).

Network PVCS — Extends Corporate PVCS for use on Networks. File locking and security levels can be tailored for each project.

PVCS for VAX systems — Requires VMS. Uses the same interface and archive format as MS-DOS version. Supports branching and offers file locking and other security features for multiple-programmer projects.

	MS-DOS*	VMS		
	PC/XT/AT	Micro VAX II	VAX 7xx	VAX 8xxx
Personal PVCS	$149			
Corporate PVCS	$395			
Network PVCS	$995**	$4,950	$9,500	$10,500+
PolyMake	$149			
Network PolyMake	$447**	$1,250	$2,375	$2,500+

*Compatible with MS-DOS 2.0 through 3.3. Compatible with the IBM PC/XT/AT & other MS-DOS PCs.

**5 Station LAN License. Call for pricing on larger Networks.

TO ORDER:
VISA/MC 1-800-547-4000.
Dept. No. 310.
Oregon & Outside USA call (503) 645-1150.

Send Checks, P.O.s to: POLYTRON Corporation, 1815 NW 169th Place, Suite 2110, Beaverton, OR 97006.

POLYTRON

High Quality Software Since 1982

CIRCLE NO. 190 ON READER SERVICE CARD

```
/* Position and then call macro #9  */
snd("\x1B*p+50X\x1B*p+70Y8\x1B&f9y3X");
snd("c\x1B&f1X");                /* End macro #8 definition */

snd("\x1B&f9Y\x1B&f0X");         /* Start macro #9 definition */
/* Position and then call macro #10 */
snd("\x1B*p+50X\x1B*p+70Y9\x1B&f10y3X");
snd("d\x1B&f1X");                /* End macro #9 definition */

snd("\x1B&f10Y\x1B&f0X");        /* Start macro #10 definition */
/* Position and then call macro #10 */
snd("\x1B*p+50X\x1B*p+70Y10\x1B&f10y3X");
snd("e\x1B&f1X");                /* End macro #10 definition */

snd("\x1B&f11Y\x1B&f0X");        /* Start macro #11 definition */
/* Position and then redefine macro #11  */
snd("\x1B*p+50X\x1B*p+70Y11");
snd("\x1B&f11Y\x1B&f0X");        /* Restart macro #11 definition */
snd("\x1B*p+50X\x1B*p+70Y11");

snd("\x1B&f11Y\x1B&f0X");        /* Restart macro #11 definition */
snd("\x1B*p+50X\x1B*p+70Y11");
snd("\x1B&f11Y\x1B&f0X");        /* Restart macro #11 definition */
snd("\x1B*p+50X\x1B*p+70Y11");
snd("f\x1B&f1X");                /* End macro #11 definition */

snd("\x1B&f-1Y\x1B&f0X");        /* Start macro #-1 definition */
/* Position and type -1  */
snd("\x1B*p+50X\x1B*p+70Y-1");
snd("g\x1B&f1X");                /* End macro #-1 definition */

/* Position cursor, execute macro #6  */
snd("\x1B*p100X\x1B*p900Y\x1B&f6y2X");

/****************** End macro depth test ****************/

snd("\x1B*p400X\x1B*p900Y\x1B&f10y2X"); /* recursive test */
snd("\x1B*p800X\x1B*p900Y\x1B&f11y2X"); /* redefinition test */
snd("\x1B*p1200X\x1B*p900Y\x1B&f-1y2X");

/* Eject page */

snd("\f");

} /* End MAIN */
```

LISTING 3: PUSHPOP.C

```
/*
 * PUSHPOP -- PC Tech Journal Laser Printer Graphics Box Test
 *
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown
 *
 * This routine uses a simple macro to draw a graphics box at 22
 * positions across the page.  A push is executed after the first 21
 * boxes are drawn.  21 pops followed by the numbers 01 thru 21
 * respectively are then printed.
 */

#include "string.h"

#define STD_OUT 1

extern void sndl(char [], int),
            snd (char []),
            setbinary(int);

/******************************** MAIN ********************************/

void main()

{
 int cnt;
 char *txt, *p;

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs
 */
setbinary(STD_OUT);

/* Initialize the printer */

snd("\x1BE");                   /* Reset the printer    */
snd("\x1B&l0O");                /* Portrait mode        */

/* Send macro defn to printer */

snd("\x1B&f1Y\x1B&f0X");        /* Start macro #1 definition */
snd("\x1B*p-95Y\x1B*p-10X");    /* Position offset           */
snd("\x1B*t150R");              /* Resolution = 150dpi       */
snd("\x1B*r1A\x0D\x0A");        /* Start raster graphic      */

/* Send the bit pattern for a box */

snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");            /* Top    */
for (cnt = 0; cnt < 31; cnt++)

   sndl("\x1B*b5W\x80\x00\x00\x00\x01", 10);    /* Mid    */
snd("\x1B*b5W\xFF\xFF\xFF\xFF\xFF");            /* Bottom */

/* End the box and the macro */

snd("\x1B*rB\x1B&f1X");

/* Position cursor, push location, execute macro */

txt = "\x1B*p??00X\x1B*p100Y\x1B&f0S\x1B&f1y2X";
p = strchr(txt, '?');           /* Point to fill location */
for (cnt = 1; cnt <= 22; cnt++)
    {
     p[0] = '0' + cnt/10;       /* Format 10s digit    */
     p[1] = '0' + cnt%10;       /* Format 1s digit     */
     snd(txt);                  /* Send to the printer */
    }

/* Pop location, write numbers */

txt = "\x0D\x0A\x1B&f1S??";
p = strchr(txt, '?');           /* Point to fill location */
for (cnt = 1; cnt <= 21; cnt++)
    {
     p[0] = '0' + cnt/10;       /* Format 10s digit    */
     p[1] = '0' + cnt%10;       /* Format 1s digit     */
     snd(txt);                  /* Send to the printer */
    }

 /* Eject page */

 snd("\f");

} /* End MAIN */
```

LISTING 4: COMPLEX.C

```
/*
 * COMPLEX -- PC Tech Journal Laser Printer Page Complexity Test
 *
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown
 *
 * This routine tests the level of complexity which the printer can
 * handle.  It draws a large number of vertical rules on the page.
 * The rules start at various vertical offsets gradually increasing
 * the complexity to beyond the limit of the HP printer.
 */

#include "io.h"
#include "string.h"
#include "fcntl.h"

#define STD_OUT 1

extern void sndl(char [], int),
            snd (char []),
            setbinary(int);
```

```
/******************************** MAIN *********************************/

void main()

{
 int cnt;
 char *txt, *p1, *p2;

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs  */

 setbinary(STD_OUT);

 /* Initialize the printer */

 snd("\x1BE");               /* Reset the printer   */
 snd("\x1B&l0O");            /* Portrait mode       */

 /* Position the cursor and draw a vertical rule */

 txt = "\x1B*p???0x!!!0Y\x1B*c1a1900b0P";
 p1 = strchr(txt, '?');        /* Point to 1st fill location */
 p2 = strchr(txt, '!');        /* Point to 2nd fill location */

 for (cnt = 1; cnt <= 109; cnt++)
     {
      /* Format the first fill location with leading zeros */
      sprintf(p1, "%03u", cnt);

      p1[3] = '0';          /* Overwrite SPRINTF terminating zero */

      sprintf(p2, "%03u", cnt);
      p2[3] = '0';          /* Overwrite SPRINTF terminating zero */

      snd(txt);             /* Send to the printer */
     }

 for (cnt = 111; cnt <= 219; cnt++)
     {
      /* Format the first fill location with leading zeros */
      sprintf(p1, "%03u", cnt);
      p1[3] = '0';          /* Overwrite SPRINTF terminating zero */

      sprintf(p2, "%03u", cnt - 110);
      p2[3] = '0';          /* Overwrite SPRINTF terminating zero */

      snd(txt);             /* Send to the printer */
     }

/* Eject page */

 snd("\f");

} /* End MAIN */
```

LISTING 5: PAGESIZE.C

```
  /*
   * PAGESIZE -- PC Tech Journal Laser Printer Page Size Test
   *
   * Version 1.0
   *
   * Copyright (c) 1987, Ziff Communications Company
   * Program by: Rainer McCown
   *
   * This program determines the printable page size by positioning
   * the cursor and then printing a character.  The HP LaserJets
   * will print the numbers through 7 (on the top, right, and bottom
   * sides) the others are off of the page.  The locations beyond
   * the left margin are negative and cannot be addressed which
   * shoves the higher numbers back onto the page.
   */


  #define STD_OUT 1

  extern void sndl(char [], int),
              snd (char []),
              setbinary(int);
```

```
/********************************* MAIN *********************************/

void main()

{

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs  */

 setbinary(STD_OUT);

 /* Initialize the printer */

 snd("\x1BE");                    /* Reset the printer   */
 snd("\x1B&l0E");                 /* Zero the top margin */
 snd("\x1B&s1C");                 /* Disable EOL wrap    */
 snd("\x1B9");                    /* Clear margins       */
 snd("\x1B&l0O");                 /* Landscape mode      */

 /* Look for top edge */

 snd("\x1B*p500x230Y@");
 snd("\x1B*p+100x-20Y1");
 snd("\x1B*p+100x-20Y2");
 snd("\x1B*p+100x-20Y3");
 snd("\x1B*p+100x-20Y4");
 snd("\x1B*p+100x-20Y5");
 snd("\x1B*p+100x-20Y6");
 snd("\x1B*p+100x-10Y7");
 snd("\x1B*p+100x-10Y8");
 snd("\x1B*p+100x-10Y9");
 snd("\x1B*p+100x-10Y0");

 /* Look for bottom edge */

 snd("\x1B*p500x3090Y#");
 snd("\x1B*p+100x+20Y1");
 snd("\x1B*p+100x+20Y2");
 snd("\x1B*p+100x+20Y3");
 snd("\x1B*p+100x+20Y4");
 snd("\x1B*p+100x+20Y5");
 snd("\x1B*p+100x+20Y6");
 snd("\x1B*p+100x+10Y7");
 snd("\x1B*p+100x+10Y8");
 snd("\x1B*p+100x+10Y9");
 snd("\x1B*p+100x+10Y0");

 /* Look for left edge */

 snd("\x1B*p130x500Y$");
 snd("\x1B*p-50x+100Y1");
 snd("\x1B*p-50x+100Y2");
 snd("\x1B*p-50x+100Y3");
 snd("\x1B*p-50x+100Y4");
 snd("\x1B*p-50x+100Y5");
 snd("\x1B*p-50x+100Y6");
 snd("\x1B*p-40x+100Y7");
 snd("\x1B*p-40x+100Y8");
 snd("\x1B*p-40x+100Y9");
 snd("\x1B*p-40x+100Y0");

 /* Look for right edge */

 snd("\x1B*p2240x500Y%");
 snd("\x1B*p-10x+100Y1");
 snd("\x1B*p-10x+100Y2");
 snd("\x1B*p-10x+100Y3");
 snd("\x1B*p-10x+100Y4");
 snd("\x1B*p-10x+100Y5");
 snd("\x1B*p-10x+100Y6");
 snd("\x1B*p-20x+100Y7");
 snd("\x1B*p-20x+100Y8");
 snd("\x1B*p-20x+100Y9");
 snd("\x1B*p-20x+100Y0");

 /* Eject page */

 snd("\f");

} /* End MAIN */
```

LISTING 6: RULES.TXT

```
/*
 * RULES -- PC Tech Journal Laser Printer Rules and Patterns Tests
 *
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown
 *
This example program was entered via a text editor and
the printer is exercised using the DOS command below:

COPY RULES/B LPT1


These first commands print all 6 of the built in patterns.

<ESC>*p100x200Y<ESC>*c300a400b1g3P
<ESC>*p500x200Y<ESC>*c300a400b2g3P
<ESC>*p900x200Y<ESC>*c300a400b3g3P
<ESC>*p1300x200Y<ESC>*c300a400b4g3P
<ESC>*p1700x200Y<ESC>*c300a400b5g3P
<ESC>*p2100x200Y<ESC>*c300a400b6g3P

The next 9 commands print all different shades of grey.

<ESC>*p100 x800Y<ESC>*c300a400b1g2P
<ESC>*p500 x800Y<ESC>*c300a400b3g2P
<ESC>*p900 x800Y<ESC>*c300a400b11g2P
<ESC>*p1300x800Y<ESC>*c300a400b21g2P
<ESC>*p1700x800Y<ESC>*c300a400b36g2P
<ESC>*p2100x800Y<ESC>*c300a400b56g2P
<ESC>*p100x1400Y<ESC>*c300a400b81g2P
<ESC>*p500x1400Y<ESC>*c300a400b100g2P

The next 4 commands print a box, a horizontal line
a vertical line, and a single dot.

<ESC>*p900x1400Y<ESC>*c300a400b0P
<ESC>*p1300x1400Y<ESC>*c300a1b0P
<ESC>*p1700x1400Y<ESC>*c1a400b0P
<ESC>*p2100x1400Y<ESC>*c1a1b0P

The last 3 boxes are printed beyond the edge of the
paper to determine the printable box margins.

<ESC>*p100X<ESC>*p-200x2000Y<ESC>*c600a400b0P
<ESC>*p2100x2000Y<ESC>*c600a400b0P
<ESC>*p1100x2000Y<ESC>*c300a1400b0P
```

LISTING 7: GRAPHICS.C

```
/* GRAPHICS -- PC Tech Journal Laser Printer Full Page Graphics Test
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown
 * This program uses the random number generator to generate
 * a random but uniformly grey backgound pattern which is used
 * to show page printing uniformity, the graphics page print
 * time, and amount of available printer memory.  To print
 * the full page requires one megabyte of available memory. */

#include "stdlib.h"
#define STD_OUT 1
extern void sndl(char [], int),
            snd (char []),
            setbinary(int);
unsigned char
  rray[7 + 85*300/10/8];    /* 7 bytes of header
                               8.5" * 300 dpi / 8bits/byte */

/******************************** MAIN ********************************/
void main()
{
 int row, cnt, hdrsiz;
 /* Change STD_OUT to binary mode to avoid converting LFs to CR,LF
    and to avoid stopping on EOFs  */
 setbinary(STD_OUT);
 /* Initialize the printer */
 snd("\x1BE");                  /* Reset the printer    */
 snd("\x1B&l0E");               /* Zero the top margin */
 snd("\x1B&s1C");               /* Disable EOL wrap    */
 snd("\x1B9");                  /* Clear margins       */
 snd("\x1B&l0O");               /* Landscape mode      */
 /* Send data to printer */
 snd("\x1B*t300R");             /* Resolution = 300dpi  */
 snd("\x1B*r1A");               /* Start raster graphic  */
 /* Send the bit pattern for a box */
 strcpy(&rray[0], "\x1B*b318W");
 hdrsiz = strlen(&rray[0]);
 for(row = 1; row <= 3300; row++)
   {
    for(cnt = hdrsiz; cnt < sizeof(rray); cnt++)
       rray[cnt] =  1 << (rand() & 7);
    sndl(rray, sizeof(rray));
   }
 /* End the raster graphic */
 snd("\x1B*rB");
 /* Eject page */
 snd("\f");
} /* End MAIN */
```

LISTING 8: RESOLUT.C

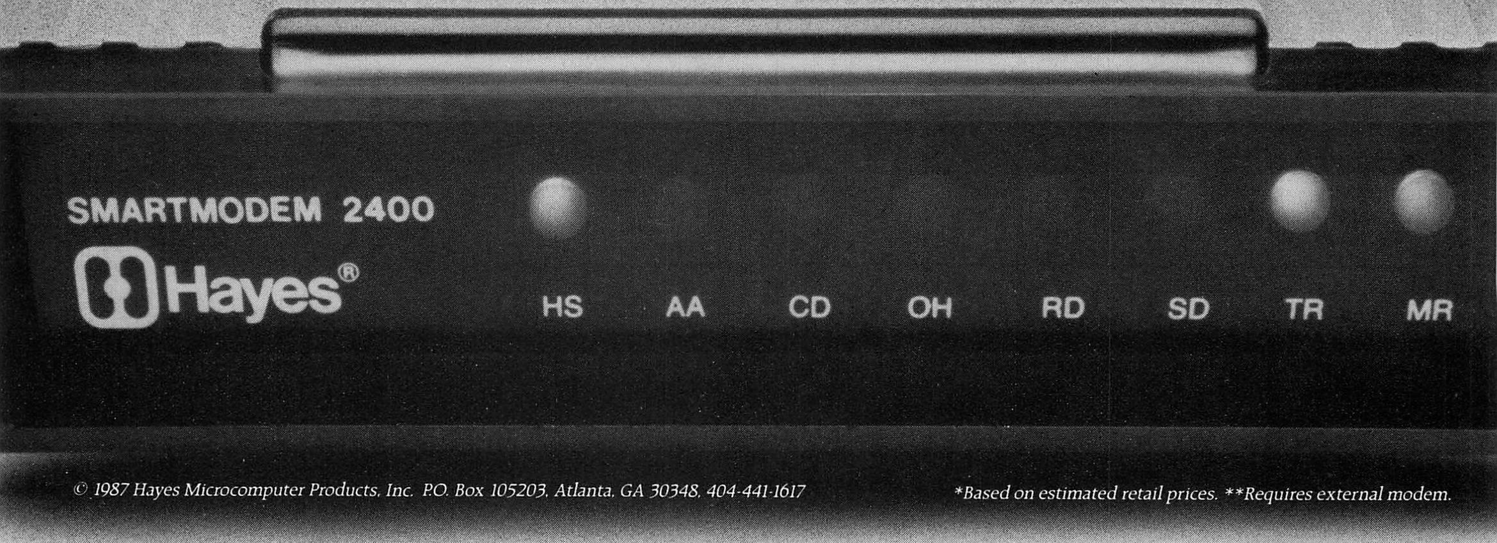
```
/* RESOLUT -- PC Tech Journal Laser Printer Resolution Test
 * Version 1.0
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown and Bob Smith
 * Prints a series of high-resolution patterns showing detailing
 * capabilities of laser printers */


#include "io.h"
#define MSIZ 64
char mat[][MSIZ+1] = {
"0000000000000001000000000000000000000000000000001000000000000000",
"0000000000000001100000000000000000000000000000011000000000000000",
"0000000000000001110000000000000000000000000000111000000000000000",
"0000000000000001111000000000000000000000000001111000000000000000",
"0000000000000001111100000000000000000000000011111000000000000000",
"0000000000000001111110000000000000000000000111111000000000000000",
"0000000000000001111111000000000000000000001111111000000000000000",
"0000000000000001111111100000000000000000011111111000000000000000",
"0000000000000001111111110000000000000000111111111000000000000000",
"1111111111111111111111111000000000000001111111111111111111111111",
"1111111111111111111111111100000000000011111111111111111111111111",
"1111111111111111111111111110000000000111111111111111111111111111",
"1111111111111111111111111111000000001111111111111111111111111111",
"1111111111111111111111111111100000011111111111111111111111111111",
"1111111111111111111111111111110000111111111111111111111111111111",
"1111111111111111111111111111111001111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111001111111111111111111111111111111",
"1111111111111111111111111111110000111111111111111111111111111111",
"1111111111111111111111111111100000011111111111111111111111111111",
"1111111111111111111111111111000000001111111111111111111111111111",
"1111111111111111111111111110000000000111111111111111111111111111",
"1111111111111111111111111100000000000011111111111111111111111111",
"1111111111111111111111111000000000000001111111111111111111111111",
"0000000000000001111111100000000000000000011111111000000000000000",
"0000000000000001111111000000000000000000001111111000000000000000",
"0000000000000001111110000000000000000000000111111000000000000000",
"0000000000000001111100000000000000000000000011111000000000000000",
"0000000000000001111000000000000000000000000001111000000000000000",
"0000000000000001110000000000000000000000000000111000000000000000",
"0000000000000001100000000000000000000000000000011000000000000000",
"0000000000000001000000000000000000000000000000001000000000000000",
};
```

NOW YOU CAN HAVE THE MODEM YOU'VE ALWAYS WANTED AT A PRICE YOU NEVER DREAMED YOU COULD GET.

At Hayes we just found a way to make the best-selling PC modems in the world even better. We lowered their price. From now on our Smartmodem 2400,™ Smartmodem 2400B,™ Smartmodem 1200,™ Smartmodem 1200B,™ Smartmodem 1200C™ and our new Smartmodem 1200A™ will cost considerably less. Up to one-third less.*

So if you've always wanted a Hayes modem, external or internal, for an IBM® PC or compatible, IBM PC Convertible, Apple® Macintosh,** Apple II, or almost any other PC, now you don't have to settle for less. Just pay less.

Hayes®



*Based on estimated retail prices. **Requires external modem.

```
char mat1[][MSIZ+1] = {
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
"0101010101010101010101010101010101010101010101010101010101010101",
};


char mat2[][MSIZ+1] = {
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
"1010101010101010101010101010101010101010101010101010101010101010",
"1110111011101110111011101110111011101110111011101110111011101110",
"0000000000000000000000000000000000000000000000000000000000000000",
"1110111011101110111011101110111011101110111011101110111011101110",
};


char mat3[][MSIZ+1] = {
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010",
"0000000000000000000000000000000000000000000000000000000000000000",
"1010101010101010101010101010101010101010101010101010101010101010";
};


char mat4[][MSIZ+1] = {
"0000000000000000000000000000000000000000000000000000000000000000",
"1111110001111100000011111000000000000000000000000000000000000000",
"0001110010001110000100011100000000000000000000000000000000000000",
"0001110100000011001000000110000000000000000000000000000000000000",
"0001111000000011110000000011100000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"0001110000000011100000000111000000000000000000000000000000000000",
"1111111110011111111100111111111000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
"0000000000000000000000000000000000000000000000000000000000000000",
};
```

```
char mat5[] [MSIZ+1] = {
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
"1111111111111111111111111111111111111111111111111111111111111111",
};

struct PRT_LINE
 {
  char lhd[5];
  char line[MSIZ/8];
 };

struct PRT_LINE
  prt_line = {'\x1B', '*', 'b', '0'+MSIZ/8, 'W'};

#define LINE_LEN sizeof(prt_line)

#define STD_OUT 1

extern void sndl(char [], int),
            snd (char []),
            setbinary(int);

/*************************** PMAT ***********************************/

void pmat(mats)

char mats[] [MSIZ+1];

{
 int row, col, bit;
 unsigned char byte;

 /* Send header info to printer */

 snd("\x1B*p+200x0100Y");       /* Position the output on the page */
 snd("\x1B*r1A");               /* Start raster graphics mode */

 /* Translate MATS into bits for output to printer */

 for (row = 0; row < sizeof(mat)/sizeof(mat[0]); row++)
   {
    for (col = 0; col < MSIZ; col += 8)
      {
       for (byte = 0, bit = 0; bit < 8; bit++)

            byte = (byte << 1) | (mats[row] [col + bit] == '1');

       prt_line.line[col >> 3] = byte;
      }

     /* Write out a line's worth */
     sndl((char *) &prt_line, LINE_LEN);
   }

 /* End raster graphics mode */
 snd("\x1B*rB");

}

/******************************** MAIN ******************************/

void main()

{
 int row, col, bit;
 unsigned char byte;

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs
  */

 setbinary(STD_OUT);

 /* Initialize the printer */

 snd("\x1BE");                  /* Reset the printer   */
 snd("\x1B&l0O");               /* Portrait mode       */
 snd("\x1B*t300R");             /* Set the printer resolution */

 /* Send the bit patterns to the printer */

 pmat(mat);
 pmat(mat1);
 pmat(mat2);
 pmat(mat3);
 pmat(mat4);
 pmat(mat5);

 /* Eject the paper */
 snd("\f");

} /* End MAIN */
```

LISTING 9: DOWNLD.C

```
/* DOWNLD -- PC Tech Journal Laser Printer Font Download Test
 *
 * Version 1.0
 *
 * Copyright (c) 1987, Ziff Communications Company
 * Program by: Rainer McCown and Bob Smith
 *
 * This routine downloads a font for a single character.  It
 * demonstrates how to construct the appropriate commands necessary
 * to define a font.  Even though the font defines but a single
 * character, it can be generalized easily.
 */

#define STD_OUT 1

extern void sndl(char [], int),
            snd (char []),
            setbinary(int);

struct FONTDESC_STR
{
 int  C26;
 char C0,
      ftype;
 int  D0,
      baseline,
      cellwidth,
      cellheight;
 char orientation,
      spacing;
 int  symbolset,
      pitch,
      height,
      E0;
 char F0,
```

```
     style,
     weight,
     typeface;
};

struct FONTDESC_STR fontdesc;

struct CHARDESC_STR
{
 char C4,
      C0,
      C14,
      C1,
      orientation,
      D0;
 int  leftoff,
      topoff,
      charwidth,
      charheight,
      deltax;
};

struct CHARDESC_STR chardesc;

char *letter[] = {
"111111000111110000000000",
"000111001000011100000000",
"000111010000001100000000",
"000111100000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"000111000000001110000000",
"111111111001111111110000",
};

/******************************** SWAP ********************************/

unsigned int swap(ibyte)

unsigned int ibyte;

{
  return(ibyte >> 8 | (ibyte & 0xff) << 8);
}

/******************************** MAIN ********************************/

void main()

{
 unsigned int msgsiz, charw, charh;
 char tmp[(50 * 50) >> 3], *p;
 int ind, i, j;

 /* Change STD_OUT to binary mode to avoid
    converting LFs to CR,LF and to avoid
    stopping on EOFs  */

 setbinary(STD_OUT);

 /* Initialize the printer */
 snd("\x1BE");                /* Reset the printer   */
 snd("\x1B&l0E");             /* Zero the top margin */
 snd("\x1B&s1C");             /* Disable EOL wrap    */
 snd("\x1B9");                /* Clear margins       */
 snd("\x1B&l0O");             /* Portrait mode       */
 snd("\x1B(8'");              /* Select symbol set Roman-8 */

 /* Define parameters of symbol set as
      0p = Fixed space
     16h = 16 cpi
      8v =  8 pt
      0s = upright
     -3b = light
      0T = line printer  */

snd("\x1B(s0p16h8v0s-3b0T");

/* Define current font ID */

snd("\x1B*c10280D");

/* Define font descriptor */

fontdesc.C26 = swap(26);       /* Constant 26 */
fontdesc.ftype = 1;            /* 8-bit (use 0 for 7-bit font
                                  type) */
fontdesc.baseline = swap(30);  /* Character baseline within cell */
fontdesc.cellwidth = swap(41); /* Width of cell */
fontdesc.cellheight = swap(42); /* Height of cell */
fontdesc.orientation = 0;      /* 0 = portrait, 1 = landscape */
fontdesc.spacing = 1;          /* 0 = fixed,    1 = proportional */
fontdesc.symbolset = swap(277); /* Using 8' for Roman-8,
                                  277 = 8*32 + 'U' - 64 */
fontdesc.pitch = swap(4*30);   /* 30 dots wide in units of
                                  quarter dots */
fontdesc.height = swap(4*50);  /* 50 dots high ... */
fontdesc.style = 0;            /* 0 = upright, 1 = italic */
fontdesc.weight = 0;          /* -7 (light) to 7 (bold) */
fontdesc.typeface =swap(7);    /* 5 = Times Roman */

snd("\x1B)s26W");
sndl((char *) &fontdesc, sizeof(fontdesc));

/* Describe the ASCII code for the letter 'n' */
snd("\x1B*c110E");

/* Define the header for the character 'n' */

chardesc.C1 = 1;               /* Constant 1 */
chardesc.C4 = 4;               /* Constant 4 */
chardesc.C14 = 14;             /* Constant 14 */
chardesc.orientation = 0;      /* 0 = portrait, 1 = landscape */
chardesc.leftoff = swap(1);    /* Within character cell */
chardesc.topoff = swap(17);    /* Within character cell */
chardesc.charwidth = swap(24); /* Width of character in dots */
chardesc.charheight = swap(18); /* Height of character in dots */
chardesc.deltax = swap(4*23);  /* Skip 18 dots after printing
                                  in units of quarter dots */

charw = swap(chardesc.charwidth);
charh = swap(chardesc.charheight);

msgsiz = sizeof(chardesc)     /* Size of header plus data portion */
       + charw/8 * charh;

sprintf(tmp, "\x1B(s%dW", msgsiz); /* Start download of character */
snd(tmp);
sndl((char *) &chardesc, sizeof(chardesc));

/* Initialize TMP to all zeros in case we re-use it */
memset(tmp, 0, sizeof(tmp));

/* Define the bits which form the character 'n' */

for(ind = i = 0; i < charh; ind += charw, i++)
for(p = letter[i], j = 0;
    j < charw;
    j++)
    tmp[(ind + j) >> 3] |= (*p++ - '0') << (7 - ((ind + j) & 7));

sndl(tmp, (charw * charh + 7) >> 3);

snd("\x1B*p342Y\x1B*p336X\x1B(10280Xn");

/* Eject page */

snd("\f");

} /* End MAIN */
```

Introducing PC AT Version 3

Ada Now

The Serious Ada Compiler For Serious Ada Programmers

If you're a serious Ada programmer today, or expect to be one—if you're serious about DoD business and those new DoD directives—you look for a lot more in your Ada compiler than just validation. And the new fourth-generation Alsys Ada compiler Version 3 for the PC AT and compatibles gives it to you.

Take code quality, for example. Version 3 generates the highest quality code of any Ada compiler on any machine! Check the PIWG benchmarks, and those of U-Michigan. Compare the quality with code from the mature C and Pascal compilers you're used to. Check especially where you might need quality most, and where Version 3 shines—in the implementation of procedure calls and exceptions. The elimination of code associated with unused subprograms gives you large reductions in code size in many applications.

Consider the Runtime Executive. True production quality. No exception-handling overhead is incurred unless an exception is actually raised. The Runtime is optimized for programs running in limited memory, or running for a long time. That's serious.

Consider the Ada-specific high level optimizer technology (not derived from Fortran or C) that removes redundant constraints checks and does so much more for code quality.

Consider robustness and reliability. Version 3 is written in Ada and bootstrapped through itself. That's proof right there that it will compile 400,000 lines of Ada code. Beyond validation, we test our compilers on hundreds of thousands of lines of extra code—from the ACEC tests, and from Ada Repository programs, and from our own specially designed code that breaks most other compilers.

There's a lot more that's serious about Version 3. The new Multi-Library environment, for example, that maximizes the efficiency of programming teams; error messages that correct, instruct, and speed the programming process; human engineering; superb documentation and customer support.

Send for our free brochure *The Many Facets of Quality.*

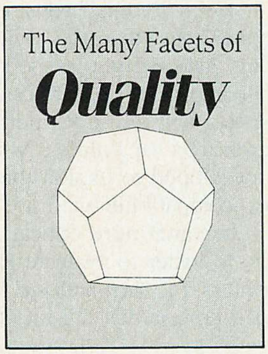
alsys

In the US: Alsys Inc., 1432 Main St., Waltham, MA 02154 Tel: (617) 890-0030
In the UK: Alsys Ltd., Partridge House, Newtown Rd., Henley-on-Thames, Oxon RG9 1EN Tel: 44 (491) 579090
In the rest of the world: Alsys SA, 29 Avenue de Versailles, 78170 La Celle St. Cloud, France Tel: 33 (1) 3918.12.44

CIRCLE NO. 105 ON READER SERVICE CARD

_____ Send me *The Many Facets of Quality.*
_____ Send me more information on the PC AT Version 3 Compiler.
_____ Call me.

Name_____
Company_____
Address_____
City_____ State_____ Zip_____
Phone_____



PCTJ 987

Alsys, Inc. • 1432 Main Street • Waltham, MA 02154

*Ada is a registered trademark of the U.S. Government (AJPO).

Laser Performance

A sampling of laser printers is measured against the Hewlett-Packard standard. Some are more compatible than others.

RAINER McCOWN and HEETH CLARK

Because they were first on the scene and because they do their jobs well, Hewlett-Packard laser printers have emerged, by intention or not, as the compatibility standard for like printers. Desktop laser-printer technology is still very young, however, so compatibility does not come easily. Manufacturers constantly strive to emulate the HP standard, but, as yet, full compatibility remains elusive.

Laser printers, typified first by the HP LaserJet, later by the LaserJet+ and LaserJet Series II, took the computer printing industry by storm because of their speed and abundant features that dot-matrix or daisy-wheel printers cannot match. With the emergence of desktop publishing, some of these features—downloadable fonts, multiple fonts per page, and graphics—are just beginning to be used.

Laser-printer hardware is more sophisticated than the available software. Although the present crop of HP-compatible printers can run it, this existing software does not fully use the hardware's capabilities. As new software is developed to exploit these capabilities, compatibility with the HP standard will become more critical.

In order to evaluate HP-compatible printers, *PC Tech Journal* has developed criteria that fall into three general categories: hardware features, timing characteristics, and performance as judged by the *PC Tech Journal* LaserJet Software Metrics (described in the previous article, "Laser Metrics," Rainer McCown and Heeth Clark, p. 74). Using these criteria, the LaserJet+ and Series II were evaluated, along with the following laser printers offering HP LaserJet+ emulation: Genicom 5010, Kyocera F-2010, Mannesmann Tally 910, Okidata LaserLine 6, QMS SmartWriter 80+, Quadram QuadLaser I, and Ricoh PC Laser 6000.

Each printer was judged for its ability to emulate the HP printer rather than for its full range of capabilities. In the testing, compatibility with the HP printer command language (PCL) was stressed, rather than similarities in laser printer hardware.

SIGNIFICANT HARDWARE

The hardware of all the desktop laser printers is very similar. Each has a control panel, cartridge slots for additional fonts, paper-handling mechanisms, and either a parallel or serial interface. The printers are generally about the same size, although newer models as well as slower models tend to be smaller.

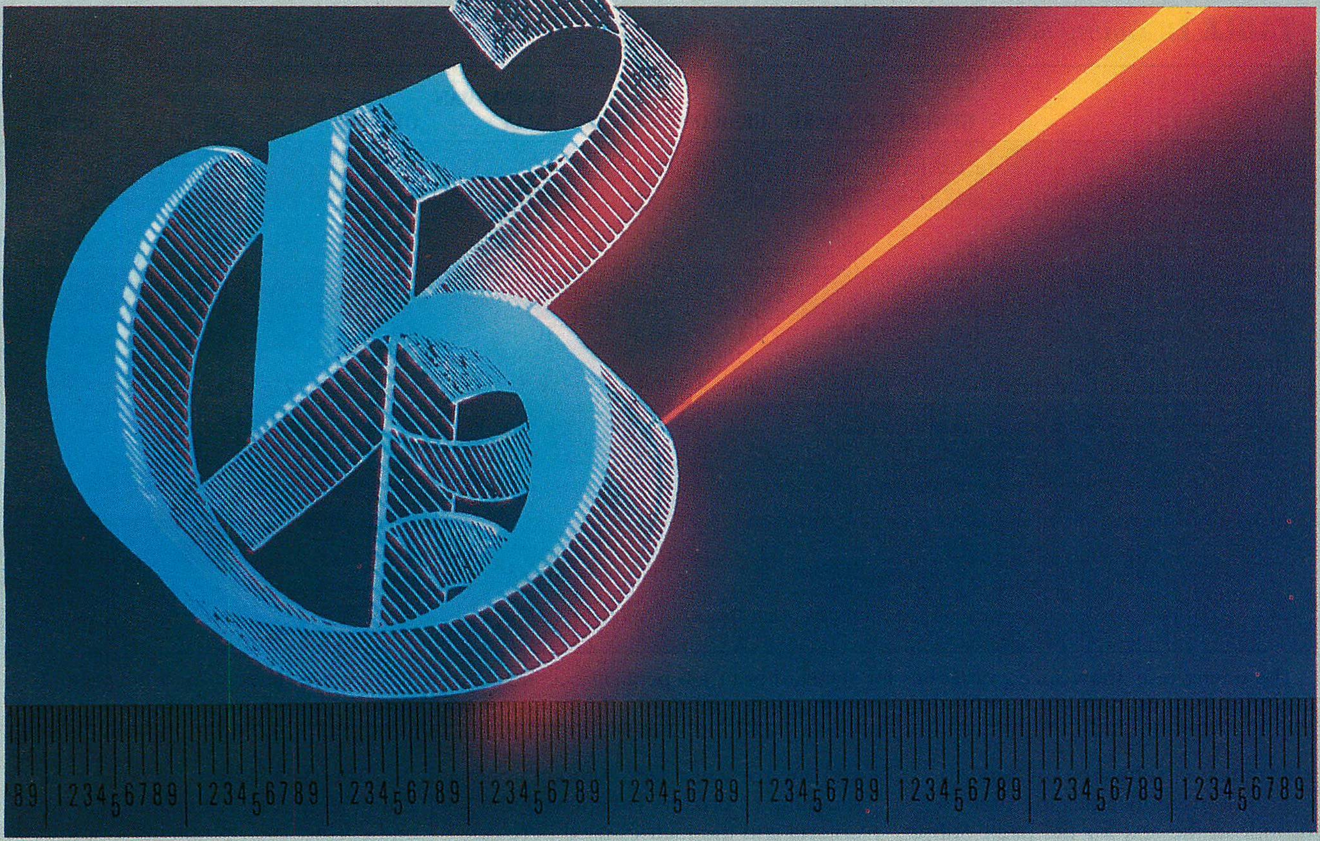
The control panel of each printer tested uses LEDs and a display to indicate status information; buttons set various options, such as *on-line* and *form feed*. Each also has lights for *on-line*, *ready*, and *data in memory*.

Some of the printers have long rows of similar indicators that make it difficult to determine current status at a glance. Some do not indicate when data are being received; this omission is especially noticeable when the printer takes several minutes to respond to a command (in a large graphic, for example).

Most of the printers provide a reset button to flush the print buffer or clear error conditions. The printers that do not have a reset switch must be turned off to flush the print buffer—which also deletes any previously downloaded fonts and resets all options to the default position.

Some have alphanumeric displays; other provide only a one- or two-character, seven-segment display. A display menu often is included if the printer has more user-selectable options than buttons. A small menu of rarely selected options can help hide some of the system's complexity; long menus can be cumbersome, such as that of the Mannesmann Tally 910, and others can be very cryptic, such as that of the Ricoh PC Laser 6000.

Most laser printers have optional hardware font cartridges and slots for



PHOTOGRAPHY • LIGHTSTRUCK STUDIOS

cartridge fonts, but not all of them can use cartridge fonts from the HP emulation mode. Because these cartridges are proprietary, the actual shape or spacing of the font characters varies from printer to printer. These individual font cartridges were not evaluated here because they are not transferable among printers. To be fully HP-compatible, an application should be able to use downloadable fonts.

Each printer's paper-handling system contains the input, printing, and output sections. The input section consists of an automatic paper-feed tray varying in capacity (from 100 to 250 sheets), paper size, and placement. Some of the printers have two automatic paper trays and most have a manual feed slot for inserting a single sheet of paper—an important feature if heavy stock or envelopes are used. Most also use a correct-order output tray with an optional straight-through paper path for heavier material.

The printing section, where toner is applied and fused to the paper, should be accessible when the cover is opened. This is where most paper jams occur. Paper quality is directly related to paper jams. When new copier paper was used in all printers tested, paper jams were either extremely rare or nonexistent. When folded or wrinkled paper was inserted, or when double-sided or multipass printing was performed, all of the printers except Quadram's QuadLaser I jammed regularly, but the jammed paper could be removed easily.

Except for the QuadLaser I, each printer tested has a straight-through paper path and manual feed tray for heavier paper. All printers provide correct-order output stackers, except those based on the original Canon CX engine, which includes both the original LaserJet and LaserJet+.

The printers vary in the complexity of their manual feeders. For example, in the HP Series II the user need only insert a single sheet, such as an envelope or viewgraph, into the manual feeder; no control-panel action is required, and the operation takes precedence over the automatic feed. Other printers are more restrictive. For example, the Kyocera F-2010 input-tray selection (which contains two input cassettes in addition to the manual feeder) cannot be changed when data are in the printer buffer. When the current tray runs out of paper, the operator cannot select the other tray or put paper into the manual input port to finish the print job. On the Genicom 5010, the user can select the manual feeder from the front panel (even when the printer has data ready), but the reset command switches the input tray back to the automatic feed tray.

Each printer uses cartridge-based engines. Those based on the Canon CX or SX engine, such as the HPs, combine the toner and the organic photoconducting cartridge (OPC), or photosensitive drum, in one unit. Printers based on other engines separate the OPC and toner; this saves the user very little cost and complicates shipping by increasing the chances for toner to spill inside the machine. Because of this danger, the Mannesmann Tally 910 manual devotes a full 12 pages to repackaging instructions.

EVALUATING COMPATIBILITY

All the printers were tested on a 6-MHz PC/AT from a parallel port. All features were related to HP emulation. A sampling of popular applications was run, including WordPerfect, Personal $\mathrm{T_EX}$'s PC $\mathrm{T_EX}$, Z-Soft's Publisher's Paintbrush, and Xerox's Ventura Publisher. In addition, each of the *PC Tech Journal* laser-printer metrics was run.

Tests were also run to measure times for warm-up and for printing one page, ten pages, and a full-page graphic. Warm-up time, from power-on until the printer is ready to print, includes any time needed to download

TABLE 1: *Timing Results*

	HEWLETT-PACKARD		GENICOM	KYOCERA	MANNESMAN TALLY	OKIDATA	QMS	QUADRAM	RICOH
Model	LaserJet+	Series II	5010	F-2010	910	Laserline 6	80+	QuadLaser I	PC Laser
Warm-up time (seconds)	64[a]	28.2	102	21	22	32	81	116[b]	35
Time to print one page of text (seconds)	23	22	23	21	20	27	21	24	25
Time to print 10 pages of text (seconds)	101	88	82	105	74	114	88	89	111
Pages per minute	6.9	8.2	9.2	6.4	10	6.2	8	8.3	6.3
Time to print full page random dot metric (seconds)	—[c]	134	268	98	—[d]	—[e]	275	Failed[f]	—[g]

Tests were performed from a diskette driven by 6-MHz IBM AT.
[a] When printer is warm, warm-up time is reduced drastically; tested at 14 seconds.
[b] Timing reflects 59 seconds warm-up plus 57 seconds to get ready for HP emulation
[c] Insufficient memory to run test on 1 page; required 3 pages. First page took 63 seconds to print; second page, 73 seconds; third page, 57 seconds. Total time was 193 seconds.
[d] Insufficient memory to run test on 1 page; required 3 pages. First page took 136 seconds to print; second page, 129 seconds; third page, 45 seconds. Total time was 310 seconds.
[e] Insufficient memory to run test on 1 page; required 2 pages. First page took 280 seconds to print; second page, 271 seconds. Total time was 551 seconds. It crashed after one-half page; had to be reset to continue.
[f] Blank page ejected after 180 seconds.
[g] Accepted full page of graphics data but printed only one-half graph. Time was 472 seconds.

The timing tests show speed variations for standard print job performance. A relatively long time is needed for a full-page graphic because each dot is defined individually when it is sent to the printer. A text image requires fewer bytes.

HP-emulation software. The time to print one page includes the time to download the text for the page. Subsequent pages are printed more quickly because the text is downloaded as the previous page is printed, and two pages can be in the paper-feed mechanism at once. The one-page and ten-page tests use fonts internal to the printer on a 356-word single-page WordPerfect text document.

Table 1 shows the maximum speed of each printer—the continuous page-per-minute rate, excluding any start-up time. In these tests, the text is transferred to the printer for each page printed; any built-in multiple-copy features are not used. This more closely approximates typical production printing; using a multiple-copy feature could produce an artificially high print rate.

The full-page graphics test is a worst-case timing test, representing the maximum time needed to download a page of data. This test is important for desktop publishing applications, which tend to be very graphics oriented.

Beginning with the two HP printers, each laser printer is described individually below. Each one is compared with the HP standard, judged on PCL emulation, and is also evaluated for ease of use, durability, and mechanical compatibility. The evaluations do not include all the specific features of each printer, but instead list the features critical to performance (including memory capacity), deviations from the HP standard, any outstanding feature, and any particular lack. Although only a sampling of printers is reviewed, the criteria and the metrics used here can be applied to other laser printers.

HEWLETT-PACKARD LASERJET+

The LaserJet+ really started all the excitement over laser printing. The follow-on printer to the LaserJet, it is based on the same Canon CX engine. The advantages of the LaserJet+ over its predecessor are internal: a defined PCL having downloadable fonts, as well as more user-available memory for these fonts and the graphics.

The Canon CX, the first desktop laser-printer engine, was developed from the Canon personal copier. Like the Canon copier, the LaserJet+ houses the OPC drum and toner supply in one cartridge; it has a 100-sheet letter-size paper tray (interchangeable with a legal-size tray, a manual-feed slot in the rear, and a right-side-up (as opposed to a correct-side-up) output-paper catcher.

The right-side-up output tray allows the operator immediately to see and correct such problems as a low toner level or dirt on the optics. However, it inconveniently delivers the last page of a document on the top and the first page at the bottom of the pile.

The control panel of the LaserJet+ has a reset switch and a two-character, seven-segment display. The *ready* light flashes as data are sent to the printer and the *form-feed* light comes on to indicate that data are in the buffer.

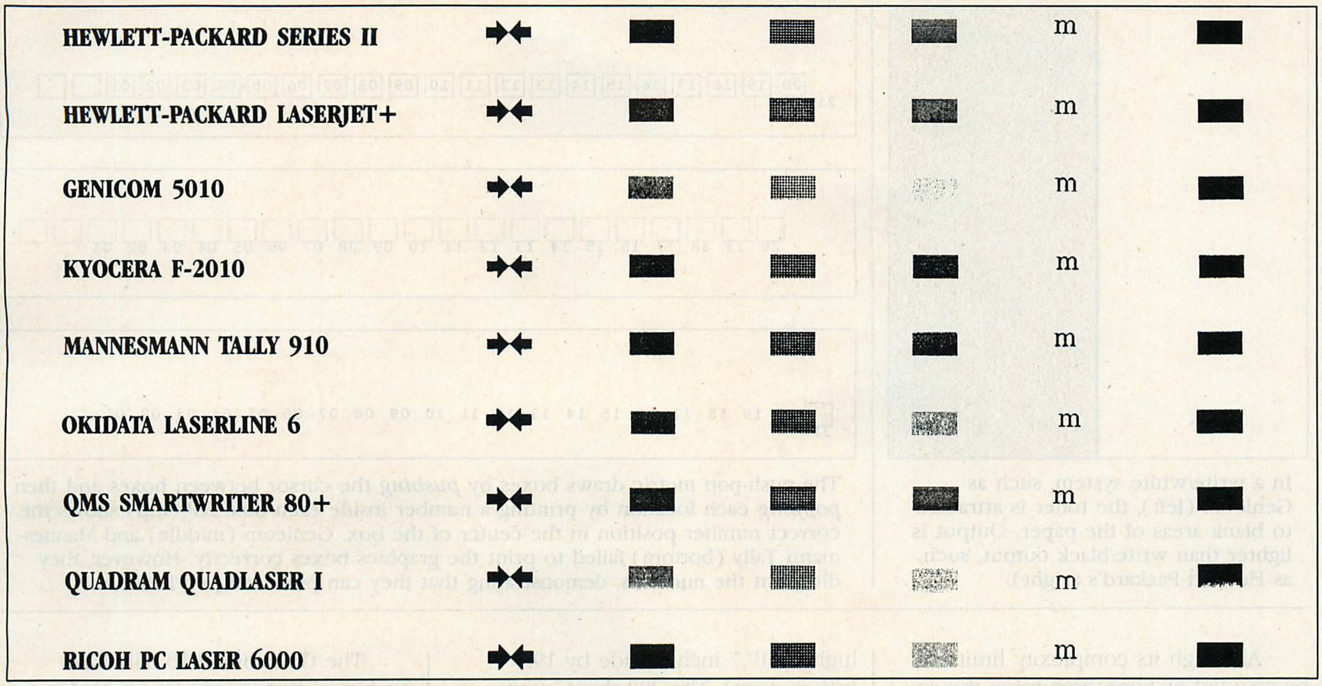
A manual input slot in the rear of the machine provides a straight-through paper path. However, the user must first select the manual feed paper option on the front panel.

Several of the tests in the *PC Tech Journal* LaserJet Software Metrics are designed to exceed the LaserJet+'s defined limits. The complexity metric, for example, was developed because the LaserJet+ had halted while printing an ordinary, although complex, document. Exceeding this limit is considered to be an asset rather than an incompatibility. The LaserJet+ and Series II pause and indicate an error message when the complexity limit is exceeded.

Another opportunity to exceed the LaserJet+ specification is in the metric testing the number of fonts allowed per page. LaserJet+ limits this number to 16; the metric was programmed arbitrarily for 19 fonts.

The macro nesting and the recursive-macro tests also provide opportunities to better the HP standard, and

FIGURE 1: *Resolution Results*

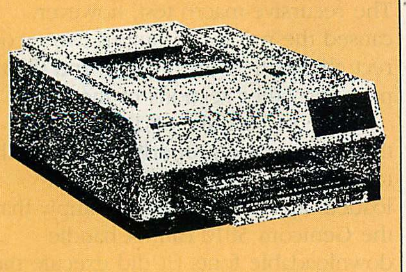


The tested laser printers show distinct variations in resolution. The ability to produce a clear image is important but must be balanced with the need to produce a shade of gray in a graphics image comparable to the shade printed by the HP machines.

many printers passed this metric. However, because this area is important only to the applications developer, a laser printer that can exceed the HP standard does not add to the standard. Applications written above this standard do not work across all of the printers but are device specific. The graphics test could not be completed on one page because of the 512KB maximum memory limitation.

The nonsoftware metrics of resolution, consistency, and black coverage are beneficial areas of improvement. The LaserJet+ performed the worst of all printers tested on the solid-black test (see figure 1).

HEWLETT-PACKARD SERIES II



Based on the new Canon SX eight-page-per-minute laser engine, the Series II is an improvement over the LaserJet+ because it is designed as a laser printer rather than as a modified copier. It is about 30-percent lighter and is shorter than the LaserJet+.

The main paper output is a correct-order paper stacker. The straight-through paper path is activated simply by opening the door at the back of the printer; and the manual paper feed slot is activated merely by inserting paper into it. The manual feed option is still available as a menu option and, if selected, inhibits the printer from accepting paper from the automatic tray.

The Series II toner cartridge also differs from the one used by the older Canon CX engine. The toner has a finer grain, allowing a higher-resolution image and blacker blacks. The cartridge is reshaped to allow a 33-percent increase in toner capacity; and the cartridge drum diameter is reduced by 44 percent, from 2.5 to about 1.1 inches. Thus, the expected life of the OPC and the toner is approximately the same, whereas in the LaserJet+, toner tended to run out before the OPC died.

This printer also departs from the typical clamshell design—the whole printer does not open in the middle. Only the center part of the top lifts for access to the toner cartridge and for purposes of clearing paper jams.

The control panel includes a 16-character, alphanumeric LCD display. The first two characters show the same two-digit status value as on the previous LaserJet models, followed by a text description of the status. A menu has been added for selecting such items as number of copies and the current font. The front panel also has a reset button; another button to print a sample of all built-in, cartridge, and downloaded (as permanent) fonts; and the same four status lights as on the previous models are present to indicate *on line, ready* (which flashes when the printer is receiving data), *manual* (paper feed), and that data are in the laser printer waiting to be printed.

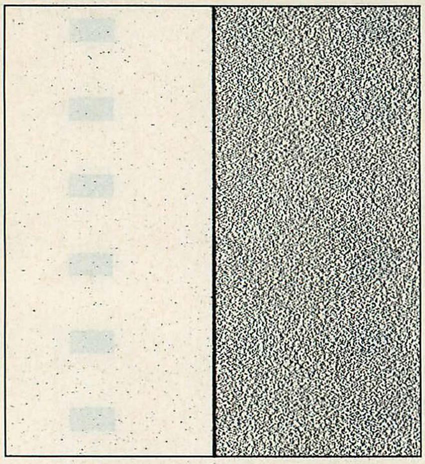
The Series II incorporates very few changes to the PCL. The only significant change is that fonts larger than 30 points can be defined for downloading.

In testing, the complexity metric failed in the same way and at the same place as the LaserJet+. Except for the full-page graphics metric, which it completed, all of the test results were identical to those for the LaserJet+.

This printer comes with 512KB of memory, but 1MB, 2MB, or 4MB expansion memory modules are available. With any of the expansion-memory modules, the Series II can produce a full-page, high-resolution graphic.

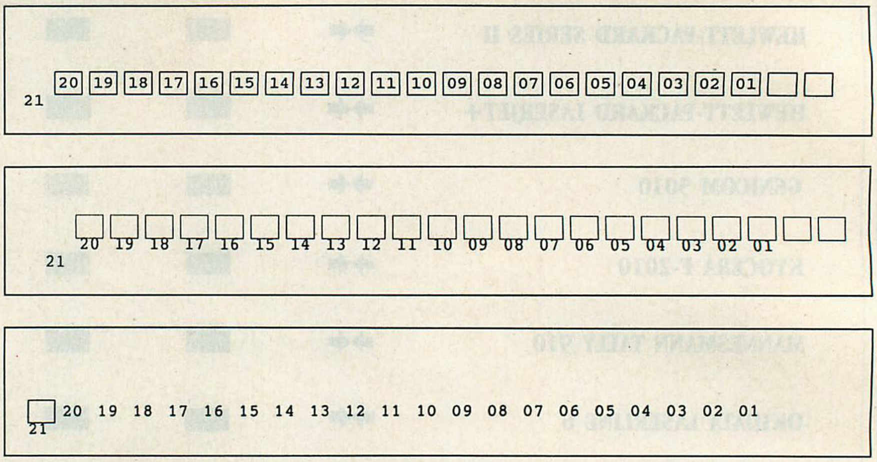
The Series II proved to be the fastest printer in many of the metrics. It was very fast on the macro, complexity, and font tests, and much faster than all the others in the full-page graphics test.

FIGURE 2: *Full-Page Graphics*



In a write/white system, such as Genicom (left), the toner is attracted to blank areas of the paper. Output is lighter than write/black output, such as Hewlett-Packard's (right).

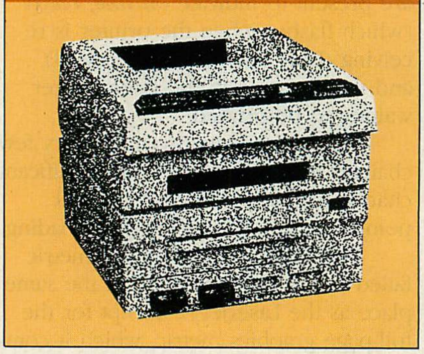
FIGURE 3: *Push-Pop Results*



The push-pop metric draws boxes by *pushing* the cursor between boxes and then *popping* each location by printing a number inside each box. HP (top) shows the correct number position in the center of the box. Genicom (middle) and Mannesmann Tally (bottom) failed to print the graphics boxes correctly. However, they did print the numbers, demonstrating that they can push and pop locations.

Although its complexity limit can be avoided by reprogramming the application on the PC, this is not always available to the user and is a nuisance. Much more preferable would be for HP to increase the number of downloadable fonts on a page from 16 to 32, as most of the compatible vendors have done, especially when the printer has 4.5MB of available memory. According to the manual, the printer can hold 32 fonts in memory.

GENICOM 5010



The 10-page-per-minute 5010 printer by Genicom Corporation is built on the Hitachi engine. When purchasing a Genicom 5010, the buyer selects a personality module that includes the printer interface. Both parallel and serial ports were provided with the review module; the active port was selected using a switch mounted on the personality module inside the machine.

The Genicom 5010 is the largest in size of any printer tested (15.2 inches high by 17.7 inches wide by 19.5 inches deep). The 200-sheet input-paper tray fits totally inside the unit from the front and handles either letter- or legal-size paper, which eliminates the need for two trays. Because it has no straight-through paper path, use of labels and heavy stock is not recommended. The manual feeder on this printer must be selected from the front panel; the reset command cancels any request for the manual feeder tray.

The control panel has a 32-character LCD display divided into two rows of 16 characters. During normal operation, seven different pieces of status information are continuously displayed. Individual buttons are used for all switchable options. The Genicom 5010 has no reset button; this lack, plus a long warm-up time, make clearing the buffer or recovering from an error rather painful for the user.

The printer has a row of very similar indicator lights, making it difficult to determine which lights are on. Among these indicators is an *active* light that shows data are in the buffer. No signal indicates when data are being transferred to the printer.

The resolution metric illustrates the write-white system (see figure 1). The single-dot line connecting the humps in the letter *m* is extremely faint if it exists at all. Although the square with the center dot missing is better defined on this printer than on any of the others tested, the individual dot pattern is very light. This is demonstrated by the full-page graphics metric output (see figure 2).

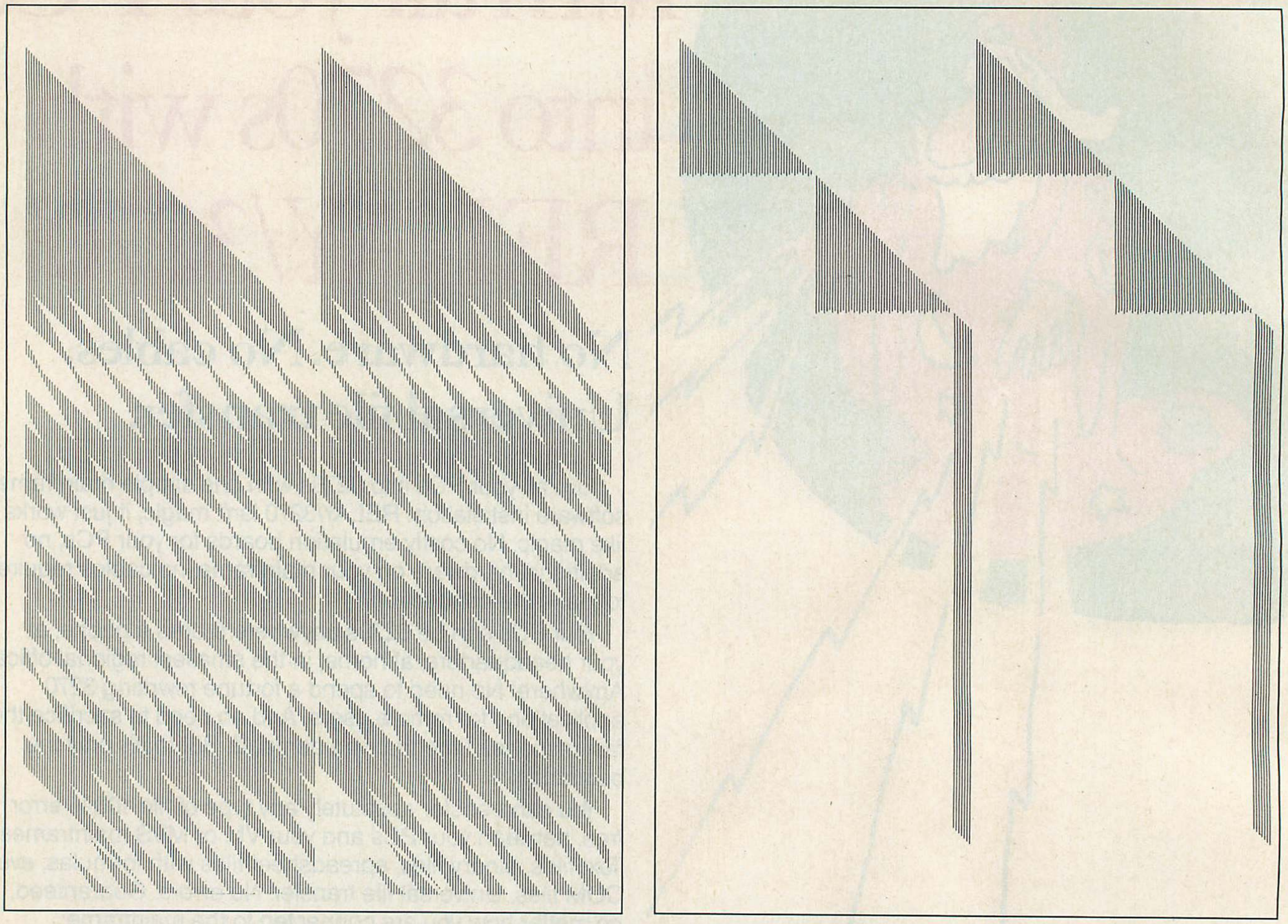
The Genicom 5010 failed the push-pop metric (see figure 3). It appeared to perform the pushes and pops, because the numbers were printed in the correct location; but it failed to combine the other commands correctly with the push-pop. In the complexity metric, the Genicom printer exceeded the HP limit and passed the metric completely when this test was run on a printer with the optional memory-expansion installed. (See figure 4 in the preceding article.)

It also failed the page-size metric. Although the positioning of the printed characters was correct, both the top and bottom margins of the page were inaccurate; the top edge of the page lost the last number while the bottom edge lost the last four numbers.

In addition, the Genicom 5010 failed the recursive-macro test. It completed the first few sections of the macro test correctly by defining and drawing the graphic macros and returning through two levels of nesting. The recursive-macro test, however, caused the printer to output two incorrect characters, eject a page, and then print the other sections of the test.

The printer also did not execute the fonts metric properly; it printed the total output in the last font downloaded. Again, this does not imply that the Genicom 5010 cannot handle downloadable fonts (it did execute the download metric properly), or that it cannot handle multiple fonts (it handled several Ventura Publisher fonts on a page). It does indicate, however, that some aspect of the font-control pro-

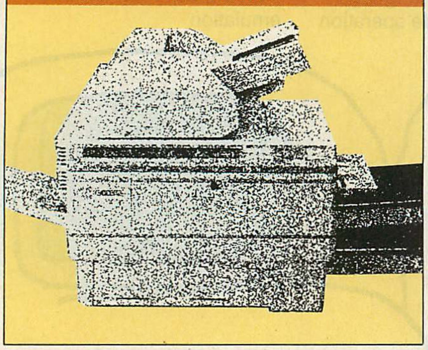
FIGURE 4: *Complexity Results*



The Kyocera F-2010's processor could not cope with a very complicated page, but failed in a different fashion (left) from the HP printers, which printed only the top portion of the design. The Mannesmann Tally 910 failed the test even earlier (right).

gram cannot handle the metric software that performs correctly on both of the HP laser printers.

KYOCERA F-2010



Kyocera uses a laser-print engine rated at 10 pages per minute in its F-2010. Our tests showed that when it prints a document in HP-emulation mode, however, the print speed drops to 6.5 pages per minute. The unit tested had 1.5MB of memory.

The F-2010 control panel has the most buttons of any printer tested. However, it has no specific indication that data are being received except that the LCD menu delivers the somewhat generic message, "Thinking"; because this message also is used for other conditions—such as off-line or data are in the buffer—the meaning is ambiguous.

Although the menu has a 32-character display of two 16-character lines, messages are rarely greater than one word. A picture of the printer on the control panel displays the current choices of input and output ports. When data are in the printer's memory, the input or output port cannot be changed, negating the advantage of the two input trays; when one tray runs out of paper, it must be refilled to finish the print job rather than simply switching to the other tray or inserting a page into the manual feed slot.

The F-2010 severely curls the paper, making multipass printing very tedious. When the temperature of the fuser roller was reduced from the factory setting, the paper still was very warm and continued to curl. Switching to the straight-through exit port slightly reduced the curling.

The toner supply lasts for about 3,000 pages of double-spaced text. In this printer, as on many of the other printers with separate OPC and toner-cartridge units, a bottle collects waste toner that bypasses the OPC mechanism. Other printers produce very little waste toner, but in the F-2010 the bottle was nearly filled. If this collection bottle is not replaced, toner could spill into the machine.

The operation of the F-2010 is also distracting. Fan noise is substantial, and when the heater on the fuser pulses on every few seconds, all of the lights in the room that are on the same circuit flicker noticeably.

The printer performed very well on the metrics, passing all except for the complexity metric (see figure 4). The processor could not keep up with the pattern and began shedding some of the load, producing an unusual pat-

NEW AND ENHANCED

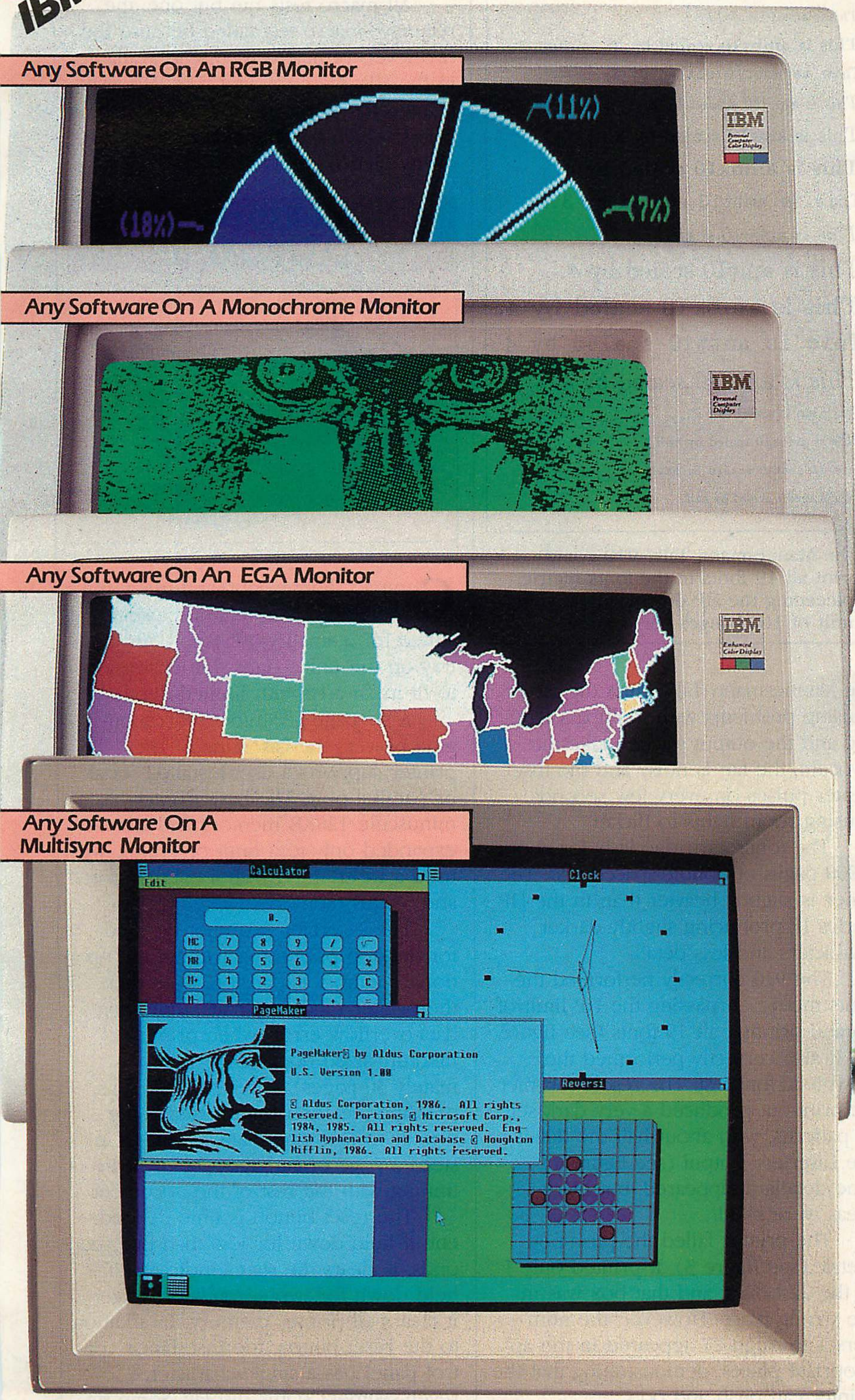
IBM VGA COMPATIBLE

EGAWONDER™

Any Software. Any Monitor. Any Time.

800x560

640x480



Now, EGA WONDER displays VGA on multi-sync monitors.

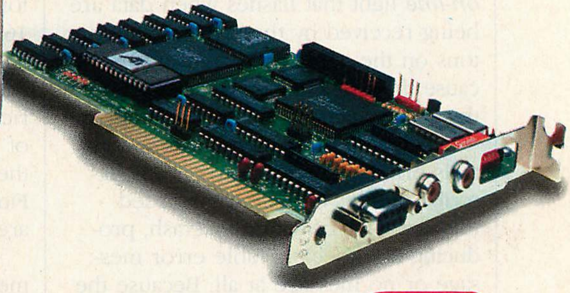
Without switching or having to take the system apart, without having to use specific monitors for specific programs, the EGA WONDER, with SoftSense Automatic Mode Switching, upgrades the EGA standard and maintains downward compatibility to older standards on any monitor.

EGA WONDER now runs the new IBM VGA[5] graphics standard plus 800×560, 752×410 and 640×480 high resolution text and graphics on MultiSync type monitors and 752×410 resolution on 25kHz 400 line monitors providing a WYSIWYG display for popular desktop publishing software such as PageMaker and Ventura Publisher. With high resolution drivers to enhance Windows, AutoCAD, GEM, 1-2-3 and Symphony, EGA WONDER displays desktop publishing, computer aided design, spreadsheets and icon operating environments with up to twice the normal resolution of EGA on MultiSync monitors.

EGA WONDER's expanded 132 column capability allows Lotus users to display a year-at-a-glance of information on 1-2-3 and Symphony spreadsheets. Connectivity support to mini and mainframe computers is provided because EGA WONDER runs SmarTerm and VTerm in 132 columns. Word processing in 132 columns is fully supported on WordStar and WordPerfect.

EGA WONDER supports 800×560, 752×410 and 640×480 modes on MultiSync monitors, 752×410 on 25kHz 400 line monitors and EGA, CGA, MDA, Hercules and 132 column modes on MultiSync, EGA, 400 line 25kHz[2], RGB[3], TTL Monochrome, Composite[3], Compaq Portable[4] and IBM PC Portable monitors.

EGA WONDER represents a new beginning for users of IBM PC/XT/AT, IBM Series 2 Model 30, or compatible computers and is now available from all major computer stores. Call us today at **(416) 756-0711** for more information.



ATI Technologies Inc.
3761 Victoria Park Ave.,
Scarborough, Ontario, Canada M1W 3S2
Tel: (416) 756-0711 Fax: (416) 756-0720
Tlx: 06-966640 (ATI TOR)

ATI® TECHNOLOGIES INC.

Technology you can Trust.

(1) Any Software, Any Monitor, Any Time applies to IBM digital monitors, graphics, standards.
(2) 132 columns not available on 25 kHz monitor.
(3) EGA, MDA, Hercules software displayed via interlacing on RGB and Composite monitors, Flickering effect of interlacing reduced with anti-glare screen filter. All other modes are non-interlaced.
(4) Via optional Compaq Expansion Module – Suggested list price – $99.
(5) Runs VGA Modes 11 & 12 on Multisync Monitors.

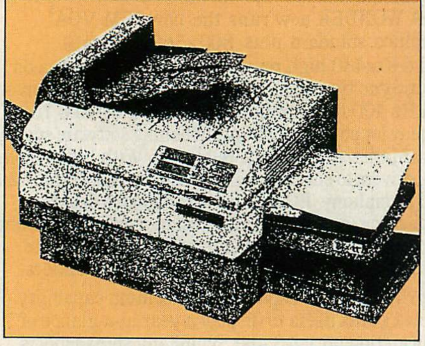
CIRCLE NO. 206 ON READER SERVICE CARD

tern of gaps in the output. The printer did not stop as commanded and the pattern continued onto the beginning of the next page. The complexity limit of the Kyocera printer was below the LaserJet+ for this metric.

In the fonts metric, the Kyocera F-2010 allowed more than the HP limit of fonts on a page. It printed all 19 fonts programmed into this metric.

This is the only printer that broke down during the testing program. Although the repair person claimed that the failure rate on this model is less than for other printers, it broke down again a couple of thousand pages later. The problem appeared to be a loose cable, which could have happened to any of the other printers.

MANNESMANN TALLY 910



The Kyocera engine also powers the 10-page-per-minute Mannesmann Tally 910. This printer can be ordered with either a serial or parallel port. The printer tested had a parallel port and 512KB of memory. The optional 1MB of memory expansion required to print full-page graphics was not tested.

The control panel consists of three buttons, a 16-character display to report printer-status and error conditions, and four lights, including a most beneficial *on-line* light that flashes when data are being received by the printer. The buttons on the panel are confusing because their functions change whenever the user selects another menu item.

A labyrinthine menu system controls printer functions. Errors are described either by a clearly worded message—or by a printer crash, producing an indecipherable error message or no message at all. Because the printer has no reset button, a crash requires a power off and on.

The 910 has a manual feed slot and two 250-sheet paper trays, which can be set to switch from one to the other when the currently used tray runs out of paper. Like the Kyocera, the Mannesmann Tally curls the paper, causing problems with multipass feeding and the output stacker. Like the Kyocera, the fan is noisy and the fuser heater pulses on every few seconds, causing room lights to flicker.

The toner cartridge lasts for about 3,000 pages of double-spaced text. The toner is slightly heavier than in the HP Series II, producing slightly darker characters and less detail.

The 910 correctly performed the fonts metric, surpassing the HP limit of 16 and printing all 19 fonts (see figure 5). It also correctly performed the download-font test. The resolution metric print-out produced a very dark set of patterns, with about half the detail of the LaserJets' output (see figure 1). Fine details disappeared, but black areas were solid.

The printer failed the push-pop metric (see figure 3). It printed only 1 of the 22 boxes, and that box was in the wrong place. However, the numbers 1 through 21 appeared in the appropriate places, demonstrating that the printer can push and pop.

The complexity metric failed, as well (see figure 4). The 910 accepted the commands until its processing capability was exceeded, which was at less than half the number of vertical lines printed by the LaserJet+. Then it purged the command buffer and continued with the next commands. The page-size metric failed on the top and bottom of the page; it printed an extra character at the top and two extra characters at the bottom.

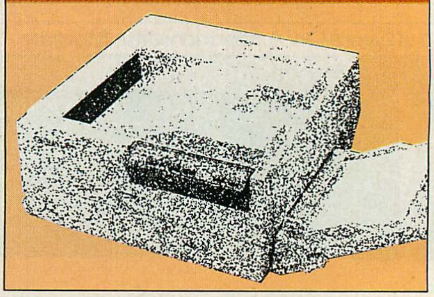
All macro tests ran but one; the recursive-macro test failed because the two levels of macro recursion available in the HPs were missing. The 910 also printed the full-scale graphic on three pages, which corresponds to its 512KB of installed memory.

FIGURE 5: *Font Results*

This is amr10 scaled by half.
This is ambx10 scaled by half.
`This is amtt10 scaled by half.`
This is amsl10 scaled by half.
This is amr10 scaled by 1.
This is ambx10 scaled by 1.
`This is amtt10 scaled by 1.`
This is amsl10 scaled by 1.
This is ambx10 scaled by 2.
`This is amtt10 scaled by 2.`
This is amsl10 scaled by 2.
This is amr10 scaled by 3.
This is ambx10 scaled by 3.
`This is amtt10 scaled by 3.`
This is amsl10 scaled by 3.
This is amr10 scaled by 4.
This is ambx10 scaled by 4.
`This is amtt10 scaled by 4.`
This is amsl10 scaled by 4.
This is amr10 scaled by half.
This is ambx10 scaled by half.
`This is amtt10 scaled by half.`
This is amsl10 scaled by half.

The Mannesmann Tally 910 could print all 19 fonts in the fonts metric, exceeding the HP downloadable font limit of 16 changes per page.

OKIDATA LASERLINE 6

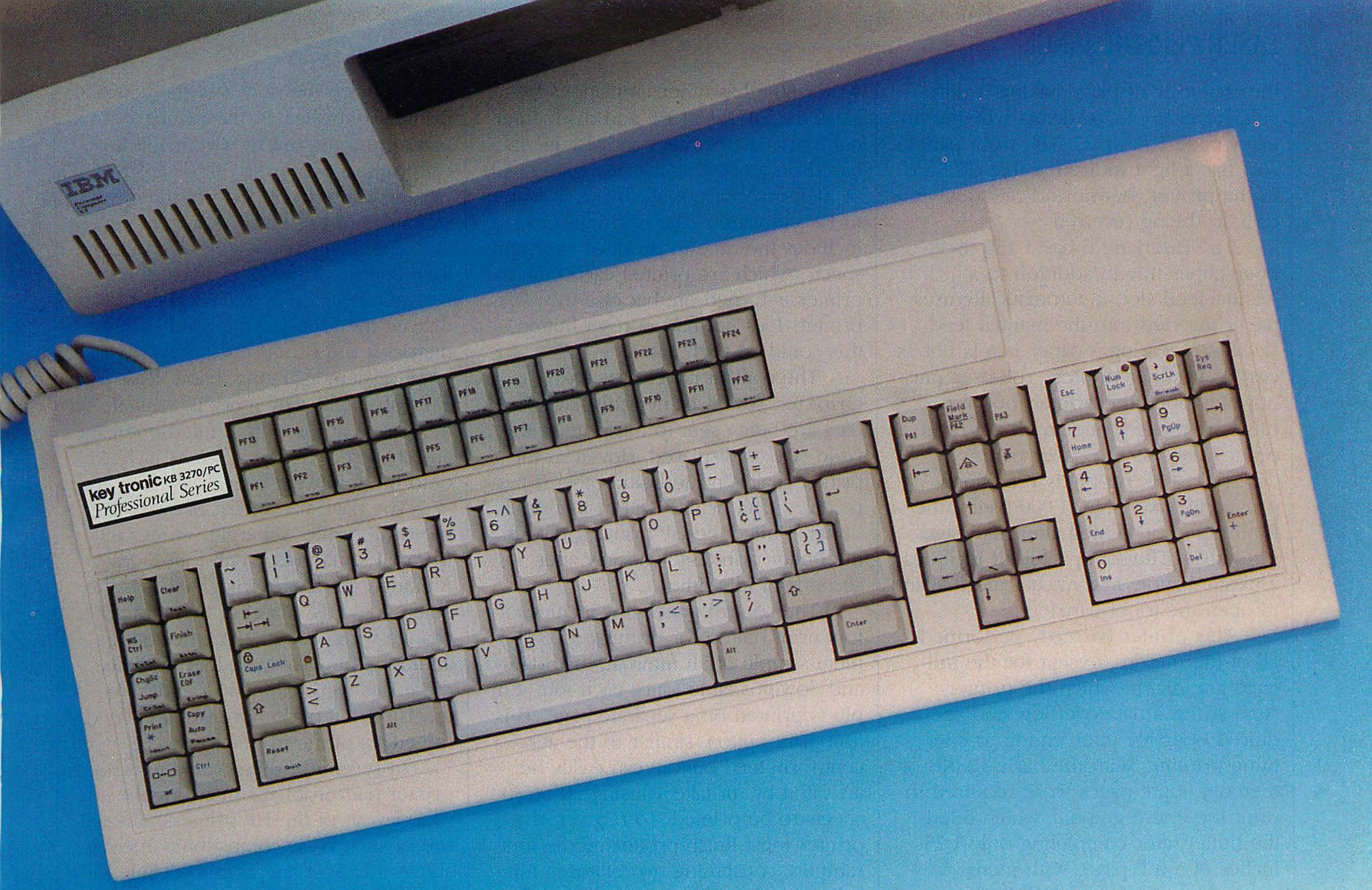


Compact, lightweight, and quiet, the LaserLine 6 is a light-duty printer suited for a small workstation. An input tray on the unit's right side enables it to fit in an area with limited depth.

A six-page-per-minute Ricoh engine powers the LaserLine 6. The printer supports a user-installed serial or parallel port. The base printer has a minuscule 128KB memory and can be expanded only to a limit of 512KB. We used a LaserLine 6 with a parallel port and 512KB memory.

The LaserLine 6 cannot be used for desktop publishing without the optional memory expansion. Even then, the printer randomly halts, indicating a data-overflow error, when printing medium-sized documents (approximately 30 pages). When this happens, pressing the reset button causes the printer to lose one page—printing instead a page of garbage—before continuing with the rest of the document.

The reset button is time dependent; if held down for less than two seconds, it clears the status indicator; if held down for two seconds or longer, it clears all printer memory. In addition to the reset button, the LaserLine's control panel has a single-character, seven-segment status display. Some status codes require two characters and are identified by sequenced flashing of



The Key Tronic® KB 3270/PC Keyboard...

The Final Step in Your Micro-to-Mainframe Connection.

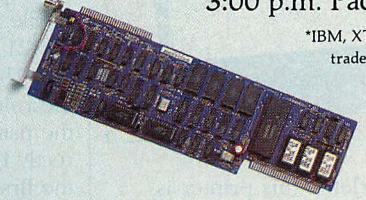
Now that you've made a major investment by allowing your micro computers to access your mainframe, take the final step by adding a Key Tronic KB 3270/PC Keyboard.

Plug compatible with the IBM* PC, XT and AT, the KB 3270/PC was developed to work in conjunction with micro-to-mainframe emulation packages from vendors such as DCA (IRMA™), CXI, FORTE' and ATTACHMATE.

By providing 122 keys including 24 function keys, the keyboard allows for easy and efficient operation resulting in more operator productivity.

It's another input solution from Key Tronic, the world's largest independent manufacturer of full travel keyboards, and a supplier of a full line of integrated input devices.

For more information about the KB 3270/PC, see your Key Tronic dealer . . . or call us toll-free at **1-800-262-6006.** (7:00 a.m. to 3:00 p.m. Pacific Time)



*IBM, XT and AT are registered trademarks of IBM Corporation.

*IRMA is a registered trademark of Digital Communications Associates, Inc.

"Feel the Key Tronic Touch"™

key tronic®

The *Responsive* Input Company™

P. O. Box 14687 • Spokane, WA 99214 • USA • (509) 928-8000 • TWX 510 773-1885

CIRCLE NO. 239 ON READER SERVICE CARD

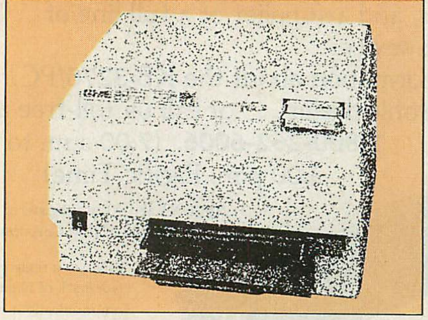
the character display, making it difficult, for example, to detect whether an error code is 2 or 22. The front panel also has a light indicating that data are in the printer, but no indication when data are being received.

The LaserLine 6 has a 150-sheet open-paper tray in addition to a manual-feed slot; it automatically uses paper inserted into the manual feed slot first. Clearing a paper jam is not as easy as in other printers; the user must not only open the printer, but also raise the hinged toner mechanism in order to access the paper path.

The toner supply is very small, providing fewer than 1,000 pages; the refill indicator lighted after 400 pages in the test. The toner-refill kit includes a plastic handle and ratchet mechanism to open the toner cartridge.

This printer correctly performed all of the metrics, except for the full-page graphic, in which the printer stopped on a data overflow and required operator intervention to continue printing. With the basic 128KB of memory, eight pages were required to print the entire graphic, which filled the buffer after completing only 1.25 inches of a full page. With memory expansion to the total 512KB, the printer completed the graphic on only two pages. The LaserJet+, with 512KB memory, required three pages to complete the graphic, indicating that more of the LaserLine's memory is directly available to the user. The fonts metric also exceeded the LaserJet+ limit of 16 fonts per page, with the LaserLine 6 completing all 19 fonts.

QMS SMARTWRITER 80+



Like the LaserJet+, this printer is based on the eight-page-per-minute Canon CX engine. This early Canon engine is rather large for its speed and lacks a correct-order output tray.

The differences between the QMS SmartWriter 80+ and the LaserJet+ are mainly in the interface electronics and the command language. The SmartWriter 80+ has other emulations besides the HP LaserJet+ and many internal fonts. The printer-control logic is based on a 68000 computer with 2.5MB of memory, but not all of it is available in HP emulation. The warm-up time includes the time to produce two status sheets, which are printed each time the printer is turned on; because the printer does not have a reset button, this could be fairly often.

The control panel uses a two-character, seven-segment display. When data are being sent to the printer, the code "PA" is displayed. However, the printer makes no indication that it has received data or has data in its print buffer. Unlike the HP printers, data in the buffer print automatically after 30 seconds of idle time.

The control panel also includes a font-print button for printing a complete sample of all internal, cartridge, and downloadable fonts; each font sample is printed on a separate page. This font-print button, similar to the adjacent form-feed button, can easily be activated by mistake, causing 20 or 30 sheets to be printed. To recover, the printer must finish producing the font samples, consuming two or three minutes; or it must be turned off and restarted, which takes only one minute but loses any downloaded fonts.

The operator may select the manual feed option from the control panel. If paper is inserted into the manual feed slot without help from the control panel, the printer not only automatically uses that paper, but also feeds a sheet from the paper tray—and the output is not always straight. For a straight-through paper path, the operator also must select the manual feed on the control panel. The SmartWriter 80+ curls the paper very little.

The resolution metric produced clean, consistent results. However, the push-pop metric failed; the printer correctly produced the boxes and the numbers inside the boxes, but the whole pattern shifted left, so the number 21 crept off the edge of the paper.

The SmartWriter 80+ also failed the rules and patterns metric; it drew the patterns and the shades of gray correctly but did not accurately position the first three blocks in the shades-of-gray test. Some of the positioning-command text printed, indicating that the printer misinterpreted the positioning commands for this section of the test. The printer was able to recover and complete the rest of the test.

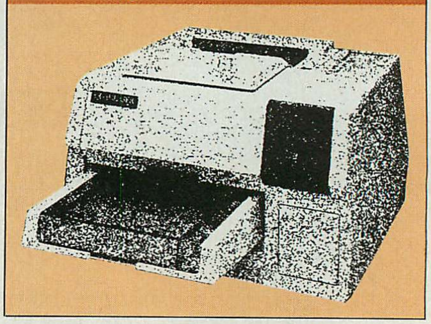
The complexity metric ran correctly to completion, which is not surprising given the power of the 68000 printer controller. In the recursive-macro test, the macro was unable to call itself to two levels, although the printer was able to perform the other examples of two-level nesting.

The SmartWriter 80+ failed the fonts metric, printing 23 lines (the correct number) in the default font rather than in the fonts downloaded by the metric. It did produce the full-page graphic, demonstrating that at least 1MB of graphics memory is available to the user, but the output filled one page and incorrectly extended 0.22 inch onto the following page.

The page-size metric showed that the QMS printer's margin settings differ from HP's margins on all four sides. Although the SmartWriter 80+ has the ability to adjust margins, they did not change when using HP emulation. In addition, the characters that straddled the edges were clipped in midcharacter. HP printers do not extend beyond the edges of the printable area.

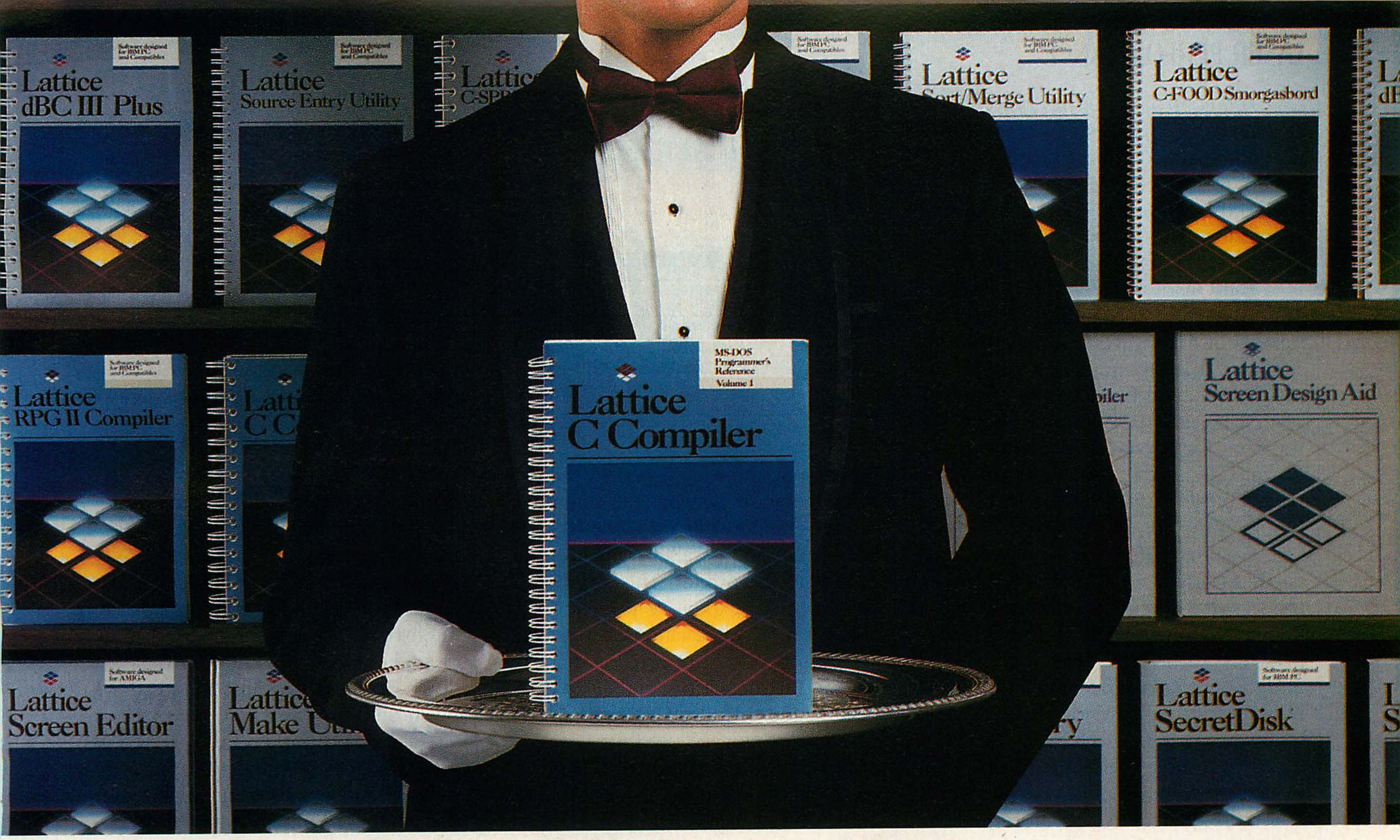
In several of these failed tests, the printer had problems matching the page layout of the HP printers. The failure of such basic compatibility metrics shows a lack of attention to detail.

QUADRAM QUADLASER I



The eight-page-per-minute QuadLaser I comes with 2MB of memory and either a serial or parallel port. It has no straight-through paper path; paper output is always in the correct-order output tray on top of the printer. It also is the only laser printer in the group tested here that has no option for using legal-size paper.

The QuadLaser I handles paper very gently, with almost no curl on the page. It also performs multipass printing consistently, without any paper jams or wrinkling. During testing, it accepted severely wrinkled paper without jamming, although the print-out was somewhat distorted by the wrinkling. None of the other tested printers handled paper as well.



Our software comes with something no one else can offer.

Lattice Service.

When you join the Lattice family of customers, you'll discover that your software purchase is backed by more than just an excellent warranty. It's backed by unparalleled technical support. By a total commitment to your success and satisfaction. And by Lattice's dedication to excellence in products and services.

Unlike other software manufacturers who charge you for services after you've purchased their product, Lattice offers a unique package of support programs at a price we can all live with—FREE.

Lattice Bulletin Board Service
LBBS is our 24-hour a day bulletin board system that allows you to obtain notification of new releases, general information on Lattice products, and programs for the serious user. And if you've ever experienced the frustration of having to wait a year or more for a new release (that has corrected a bug), you'll really appreciate LBBS. Because with this service, you can actually download the latest program fixes to instantly eliminate any bugs discovered after release.

Available through dealers and distributors worldwide.

Technical Support Hotline
Responsible, dependable and capable Support Representatives are only a phone call away. You will talk to a highly skilled expert who is trained to answer any questions you have relating to specific Lattice products. Remember, your complete satisfaction is our goal.

McGraw-Hill BIX™ Network
The Byte Information Exchange (BIX) Network is a dial-in conference system that connects you with a Special Interest Group of Lattice users. The nominal one-time registration fee allows you to BIX-mail your questions—via your modem—directly to Lattice. Or you can post your questions in the conference mode for Lattice or other users to answer. Once again, you have 24-hour access.

You Also Receive:
- Timely updates and exciting enhancements
- 30-day, money-back guarantee
- Lattice Works Newsletter
- Technical Bulletins
- Access to Lattice User Groups

Lattice has developed more than 50 different Microcomputer software tools that are used by programmers worldwide. We were there for every MS-DOS release. We're there now for OS/2. And we'll be there for the next generation of technical changes. But most of all, Lattice is there for you.

Lattice
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600

CIRCLE NO. 160 ON READER SERVICE CARD

The control panel has only one switch on the front, which is used to go on-line and off-line as well as to issue a form feed. The latter is achieved by pressing it twice in quick succession. The status display consists of one digit plus several symbols.

The QuadLaser's lack of a reset button is a real shortcoming because turning off the power to reset means reloading the HP emulation and all of the downloaded fonts, which can take several minutes. The printer also has no font-cartridge slots; all fonts must be downloaded, except for the built-in 10-pitch Courier font, which is inaccessible from the HP emulation.

Although future versions will have hardware fonts, QuadLaser's HP emulator software currently resides on the PC and is downloaded each time the printer is turned on. Due to a large input buffer, accidentally printing a binary file can eject hundreds of pages unless the printer is reset.

The printer is designed to be controlled from the PC and provides several utility programs and more than 140 downloadable fonts in two primary type-style families: Times Roman and Helvetica. Each style is provided in 10 sizes (6 through 36 points) and three weights: normal, bold, and italic.

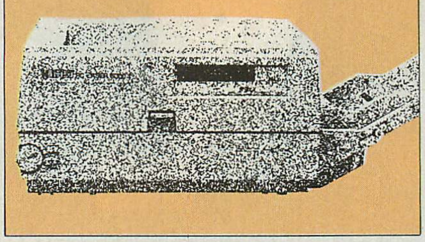
The Ricoh engine of the QuadLaser I does not use the typical clamshell design for accessibility; instead, the OPC and toner are loaded from the front through an insecurely suspended drawer. While operating, the printer sounds like a loose conveyer belt.

This printer executed both the push-pop and the fonts metrics. However, the complexity metric failed—the QuadLaser I printed nothing at all.

The page-size metric produced correct output on the top and sides but the bottom limit was set too large, with three extra digits printed at the bottom of the page. The rules and patterns metric failed because the QuadLaser I did not print the single-dot-width vertical-rule test or the single-dot rule test. The blacks printed very dark, even, and solid. In the macro metric, the QuadLaser I allowed more than two levels of macro nesting, did not permit the two levels of macro recursion allowed by HP, and completely failed to execute the macro-redefinition test.

The full-page graphic metric printed only a blank sheet; and the full-page Mandelbrot printed very lightly. These results show that although the printer is capable of printing full-page graphics, it has trouble with an oversize picture.

RICOH PC LASER 6000



Ricoh Systems uses its own second-generation, six-page-per-minute printer engine in its PC Laser 6000. Both serial and parallel ports are standard with the printer. This small, lightweight, and inexpensive printer is suitable for a PC workstation. It is not designed for long print runs or shared use, and it runs at a slightly slower print speed than the HP printers.

The printer we used for testing purposes had the standard 512KB of memory; the memory expansion up to 2MB was not tested. The printer was evaluated with its optional LaserJet+ emulation card. It stopped functioning after about 1.5 hours of continuous use, and then began working again normally after a rest period, except for a developed squeak.

The control panel has a two-digit display that makes changing modes and selecting front-panel options rather cryptic. The user must consult the manual to decipher options and settings that are identified only by numerical codes on the menu. Setting the printer to default to the HP-emulation mode requires a key sequence that is not described in the manual, but was supplied by Ricoh's technical support group. The control panel includes a reset button and a data-entry light, both of which are major conveniences.

The PC Laser 6000 has a 150-sheet, uncovered input-paper tray that can be refilled without being removed. The tray can handle letter- or legal-size paper, either of which is selected by a knob on the front panel of the printer. A sheet of paper inserted into the manual feed slot takes precedence over the automatic feed tray.

The toner is rated for 1,500 sheets (750 sheets the first time). On the tested model, the original toner supply ran low after 400 sheets, turning on the replenish-toner light.

The PC Laser 6000 failed the macro, page-size, and the full-page graphic tests. For the full-page metric, the printer accepted all of the data but, due to lack of memory, printed only a half-page of the data and discarded the rest. This is much less desirable than printing the information on several pages. The other printers with memory limitations that we tested completed printing the data on several pages.

The output for the resolution metric is shown in figure 1. Fine lines are clear and crisp, but in the uniform, high-resolution patterns, a good deal of printer-induced variation is evident.

PRINTER VARIATIONS

Because of the large variety of laser printers available, selecting the best one is a compromise based on the user's particular requirements. This evaluation was made for a technical application, with a bias toward desktop publishing. It was limited to a sampling of printers that provide HP emulation—of paramount importance to sophisticated users.

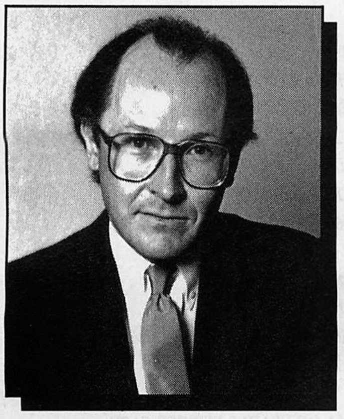
Durability varies among the printers tested. The HP printers appear durable enough to support medium-duty use among a limited group of users. The Okidata LaserLine 6 and the Ricoh PC 6000, both based on the six-page-per-minute Ricoh engine, have markedly lighter construction and smaller toner capacity, limiting their use to light duty only. The Genicom 5010 is at the other extreme; it supports heavy-duty use among many individuals.

As for mechanical capability, all the printers tested, except the Quadram QuadLaser I, have a manual feed slot and can handle legal-size paper. The HP Series II, Ricoh PC Laser 6000, and Okidata LaserLine 6 all default to the manual feed slot when paper is inserted. Without this capability, the operator must select the manual feed option from the control panel.

The reset button and data-entry indicator are important controls not available on all the laser printers. The Genicom, Kyocera, Mannesmann Tally, QMS, and Quadram printers do not have a reset button—a substantial failing because situations that require resetting the printer happen frequently in a development environment. This omission is especially problematic with the Quadram QuadLaser I printer because the HP emulation is software that must be reloaded from DOS whenever the printer is turned on.

Although every printer tested has a light to indicate that the printer buffer contains data waiting to be printed, many of the printers also have a light that flashes when data are being accepted by the printer. This is an impor-

GEM®: 3 YEARS AND GOING STRONG



We first released our Graphics Environment Manager™ (GEM) software three years ago and it's time to let you know what we have achieved. I also want to thank the software developers who have selected from Digital Research® a solution for their graphics programming needs. I extend a personal invitation to all software developers to join the group of over 200 software vendors shipping GEM software compatible products. GEM provides an easy approach to the integration of graphics within applications while maintaining the speed and performance that today's customers demand.

Dick Williams
President
Digital Research Inc.

GEM: POWER, PERFORMANCE AND PROGRAMMER FRIENDLY

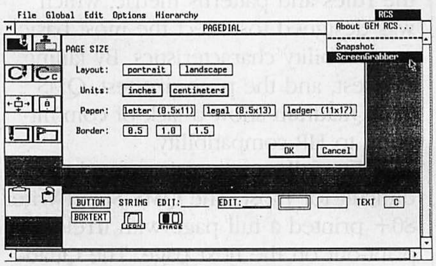
GEM: Setting The Standard For Desktop Publishing

What do the top Desktop Publishing applications have in common? GEM! GEM provides a software developer with the ability to concentrate on the features and functionality of his application while leaving support of graphics hardware to a mature full-featured virtual device interface, the GEM VDI. And there's more! The GEM Application Environment Services (AES) provides the ability to integrate windows and drop down menus into your user interface with a minimum of effort. That's why the FIRST, the LATEST, and the BEST desktop publishing applications sold for IBM® PC's are all GEM-based products. Including:

- DAWN™ from Compugraphic®
- DESIGNER™ from Archetype
- GEM® DESKTOP PUBLISHER™ from Digital Research Inc.
- PAGEmaster™ from Advanced Vision Research, Inc.
- THE OFFICE PUBLISHER® from Laser Friendly, Inc
- VENTURA PUBLISHER™ EDITION from Xerox®

GEM: Extensive Driver Support

Software analysts see the emerging graphics application market as continuing its upwardly spiraling growth. Support of the latest devices is key to any application seeking to make an impact in this highly competitive area. The key to your success is DRI's commitment to the support of the latest technology. GEM supports over 150 devices including hi-res displays, laserwriters, postscript, dedicated graphics processors, scanners and metafiles. That's one of the reasons McDonnell Douglas chose GEM as the graphics environment for its CAD package CROSSROADS™ and Personal CAD Systems® chose GEM for PC-CAPS®. More than 15 CAD/CAM applications have been developed under GEM because GEM PROVIDES HIGH PERFORMANCE SOLUTIONS TO PROFESSIONAL PROGRAMMERS.



GEM Programmer's Toolkit: Tools That Work with Any Compiler

GEM allows you to work with any popular compiler you choose. GEM applications have been written in MODULA II, C, PASCAL, FORTRAN, ADA, TURBO PASCAL and TURBO C. The GEM® Programmer's Toolkit™ provides a source level sample set of bindings, complete documentation, a utility to help you visually build your user interface and more. If you need help, contact our personalized software developer support program.

Over 1.4 Million Installed GEM Software Customers and Growing Rapidly.

There are over 200 GEM applications shipping for the IBM PC and compatibles, and more than 700 applications available for the ATARI® ST™. GEM PROVIDES SPEED AND PERFORMANCE ON 8088/8086s THROUGH TO 80386s. The programs that you develop today will run on your customers' IBM PC/XT's, AT's, IBM Personal System/2™ microcomputers and compatibles. GEM applications released over two years ago for the IBM PC work today on the Personal System/2 with no modifications. Further, Digital Research is committed to providing support of a consistent Application Programming Interface under future environments. Your programmers will not have to rewrite major portions of their code as you reach out to support newly emerging standards.

GEM: The Programmer's Solution

Join the ranks of successful software developers who have already released GEM applications. Contact DIGITAL RESEARCH or your distributor for additional information on the GEM PROGRAMMER'S TOOLKIT, available at a retail list price of only $500.

Call 1-800-443-4200

DIGITAL RESEARCH®

CIRCLE NO. 220 ON READER SERVICE CARD

tant indication for large graphic data files that take several minutes to transfer, as well as for applications, such as desktop publishing, that take considerable time to format before data are sent to the printer.

Speed varies among printers, with the HP Series II performing best overall. The greatest variations in timing were in the pages-per-minute and the graphics tests. We found that the measured speeds of the printers are close to the manufacturers' rated speeds, except for the Kyocera, which performed below its rating (6.4 pages per minute rather than the rated 10).

The HP LaserJet+ and Series II have 512KB of memory, which is the minimum required to run desktop-publishing applications. The Series II can be expanded to 4.5MB of memory. The Okidata has 128KB memory but can be expanded to 512KB. All the others are available with between 512KB and 2.5MB of memory.

All of the printers will certainly run any of the current crop of applications software. However, more sophisticated laser-printer features reveal dramatic differences. All the manufacturers of the tested printers claim HP emulation, but the *PC Tech Journal* LaserJet Software Metrics show that all have some level of incompatibility.

Some of the laser printers outperform the HP standard in the fonts and complexity metrics. The number of fonts allowed per page is an important consideration because each change in size (such as for footnotes or titles) or weight (for bold or italic) constitutes a new font. The HP's limit is 16 fonts per page. The Kyocera, Mannesmann Tally, Okidata, Quadram, and Ricoh printers correctly printed all 19 fonts included as part of the fonts metric—and probably could have printed more. The Genicom 5010 printed all of the lines of text using the last font downloaded, and the QMS SmartWriter 80+ printed only one font, which was not even one of the downloaded fonts.

In measuring complexity, the Genicom, Okidata, and QMS printers came through admirably, all completing the vertical line design demanded by the complexity metric. This metric deliberately exceeds the HP limit of complexity to show the capabilities of compatible printers. The HP printers failed this test by not completing the design. The Kyocera F-2010 not only printed the design incorrectly, but then did not flush the buffer and printed trash on the following page. The Mannesmann Tally 910 completed less than half of the design, and the Quadram QuadLaser I failed to print anything at all. The Ricoh printer also failed to complete the pattern correctly.

All printers passed the resolution metric, but substantial variations were noted among the printers.

The macro metric spelled doom for the Genicom, Mannesmann Tally, Quadram, and Ricoh printers. Ricoh's PC Laser 6000 got caught in the recursive macro and continued printing pages until it was turned off manually. When a printer fails this kind of test, it should be turned off and restarted to ensure that the printer does not remain in an unstable condition.

The push-pop metric was unsuccessful on the Genicom, Mannesmann Tally, QMS, and Quadram printers. The Genicom and the Mannesmann Tally printers performed the push-pop instructions correctly but failed the metric because of the interaction with other commands in the tests.

The Genicom, Mannesmann Tally, QMS, and Quadram, and Ricoh printers could not handle the page-size metric. A lack of attention to details, rather than any fault of the printer engines, appeared to cause these failures.

QMS and Quadram printers failed the rules and patterns metric, which was designed to reflect the most basic compatibility characteristics. By failing this test, and the page-size test, QMS and Quadram show a lack of commitment to HP compatibility.

The full-page metric proved too difficult for most. The QMS SmartWriter 80+ printed a full page with irrelevant print-out on the next page. The Quadram QuadLaser I did not print anything at all. The Ricoh PC Laser 6000 printed only half a page, discarding the rest. The Okidata LaserLine 6, Mannesmann Tally 910, and HP LaserJet+ took several pages to complete the print-out, but were able to print all of the data. The Genicom, HP Series II, and Kyocera printers had enough memory to print the correct output.

HP has the most experience producing desktop laser printers, which it has used to improve its product, producing a true second-generation laser printer in the Series II. HP carefully evaluated form, fit, and function to produce a superior product.

Successfully using the full capacities of a laser printer requires a long learning curve and much trial and error. Only with practice and patience can the operator reach the level of sophistication needed to produce documents of near typeset quality.

Hewlett-Packard
3000 Hanover Street
Palo Alto, CA 94304
800/752-0900
LaserJet+: $3,995
Series II: $2,595
CIRCLE 360 ON READER SERVICE CARD

Genicom
Genicom Drive
Waynesboro, VA 22980-1999
800/437-7468; 703/949-1170
5010: $3,695
CIRCLE 361 ON READER SERVICE CARD

Kyocera
3165 Adeline Street
Berkeley, CA 94073
800/334-6056; 415/848-6680
F-2010: $4,695
CIRCLE 362 ON READER SERVICE CARD

Mannesmann Tally
8301 S. 180 Street
Kent, WA 98032
800/843-1347; 206/251-5500
910: $3,695
CIRCLE 363 ON READER SERVICE CARD

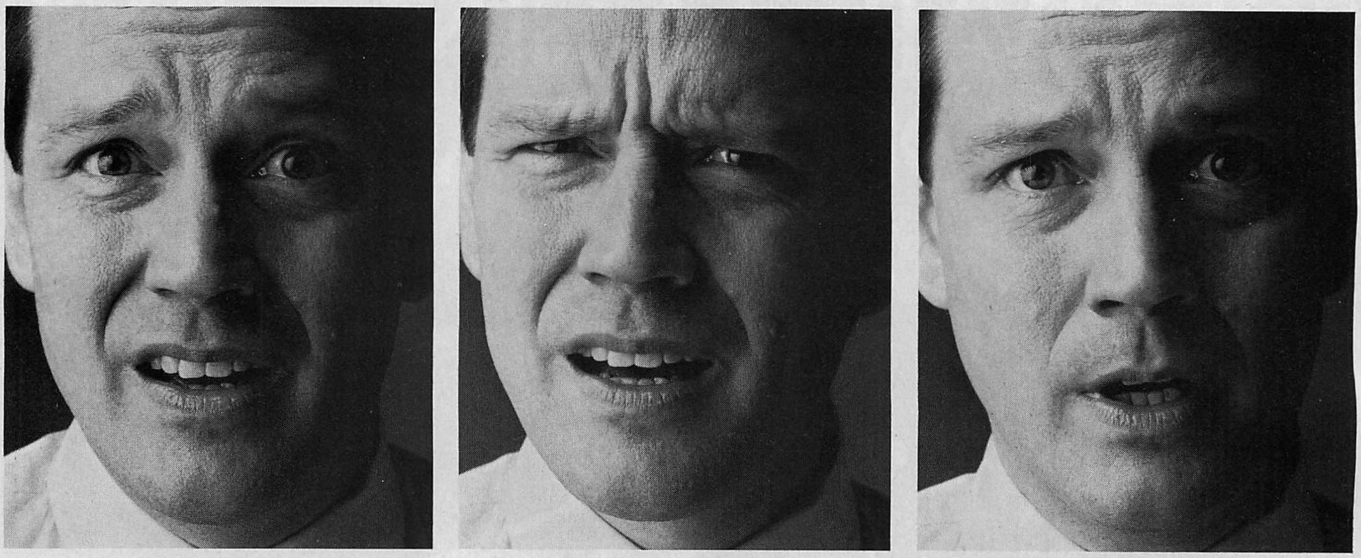
Okidata
532 Fellowship Road
Mount Laurel, NJ 08054
800/654-3282;
609/235-2600
LaserLine 6 (with HP compatibility module): $1,995
CIRCLE 364 ON READER SERVICE CARD

QMS
One Magnum Pass
Mobile, AL 36618
205/633-4300
SmartWriter 80+: $4,595
CIRCLE 365 ON READER SERVICE CARD

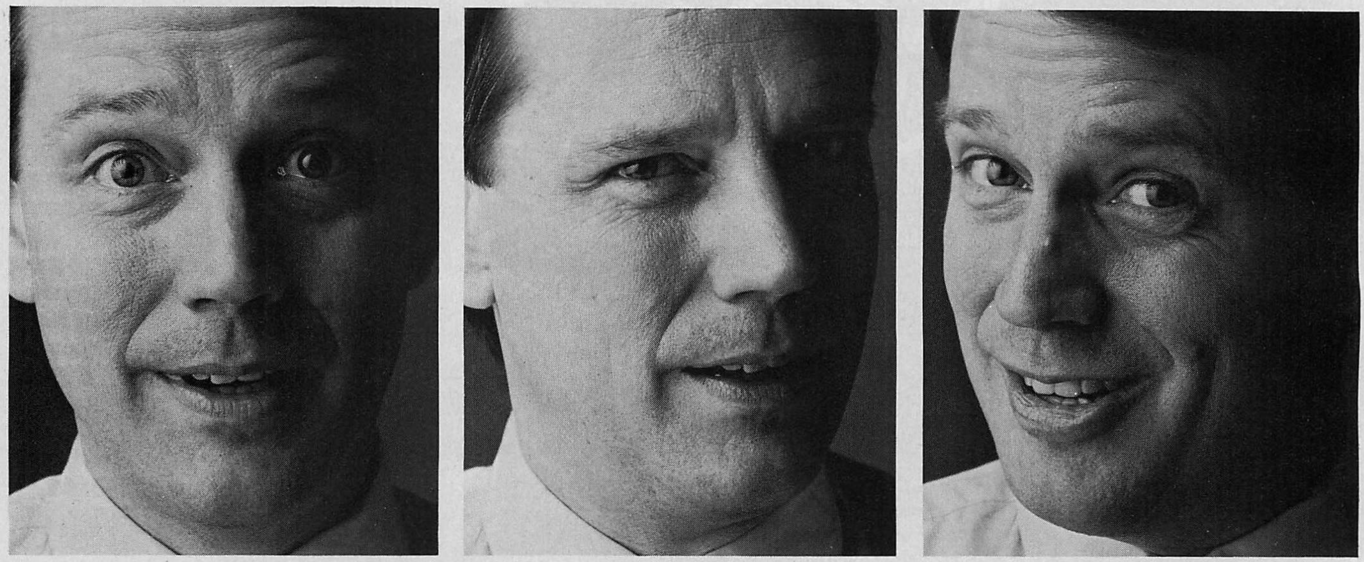
Quadram
One Quad Way
Norcross, GA 30093-2919
404/564-5566
QuadLaser I: $3,795
CIRCLE 366 ON READER SERVICE CARD

Ricoh Systems
5 Dedrick Place
West Caldwell, NJ 07006
201/882-2000
PC Laser 6000: $2,395
CIRCLE 367 ON READER SERVICE CARD

Rainer McCown is president of Rhintek, Inc., a systems software company that specializes in developing products and consulting in the areas of communications, compilers, editors, and graphics. Heeth Clark is an instructor in computer literacy at The Johns Hopkins University in Baltimore.



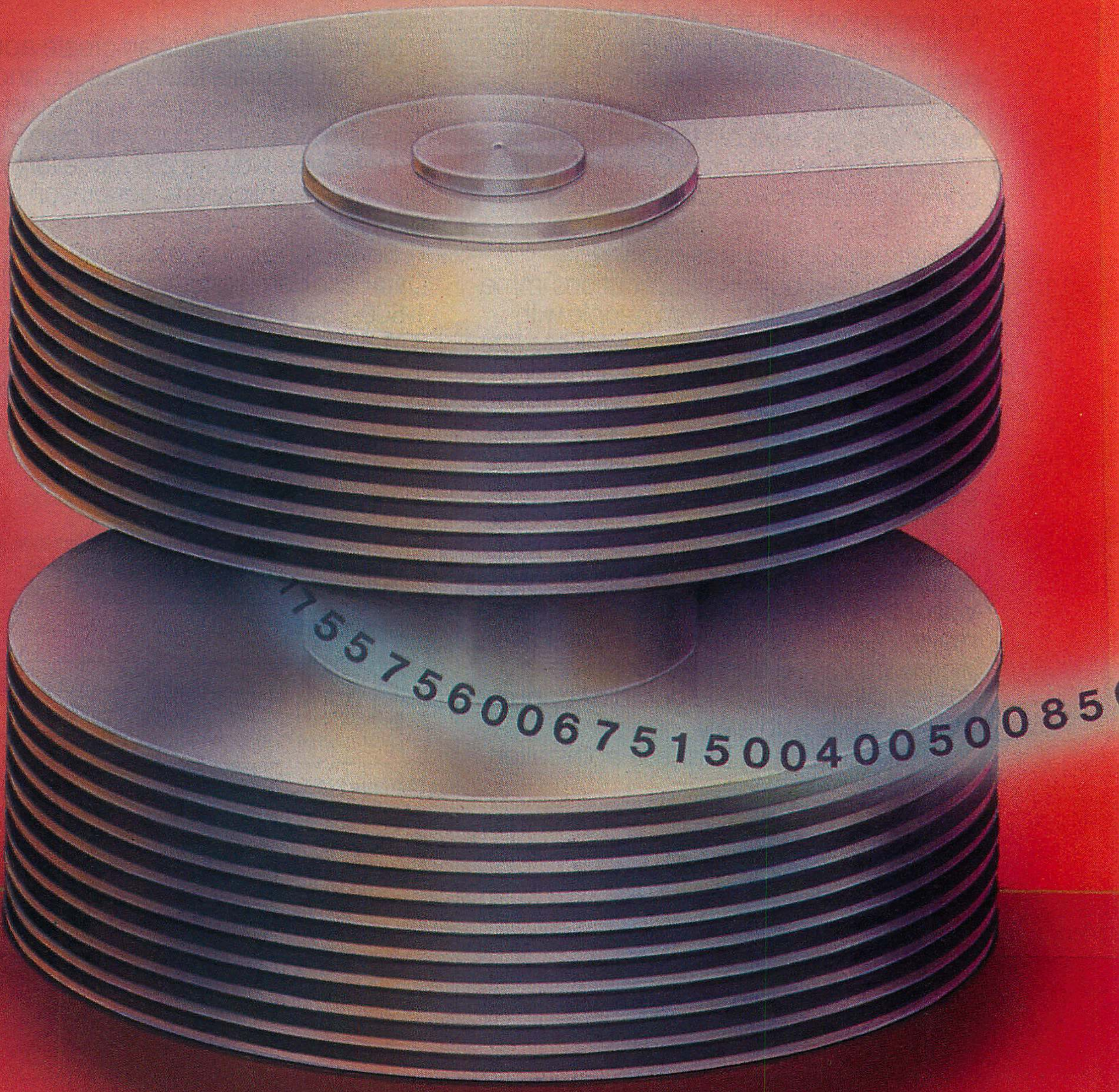
1:24: *"They want the report in the morning. I'll be here all night."*



3:47: *"This is going too fast. I must be forgetting something."*

5:05: *"Having saved the day again, our hero rides off into the sunset."*

Refining Mainframe Access



CDI's AutoKey/3270 improves the user interface and simplifies communications in a PC-mainframe environment.

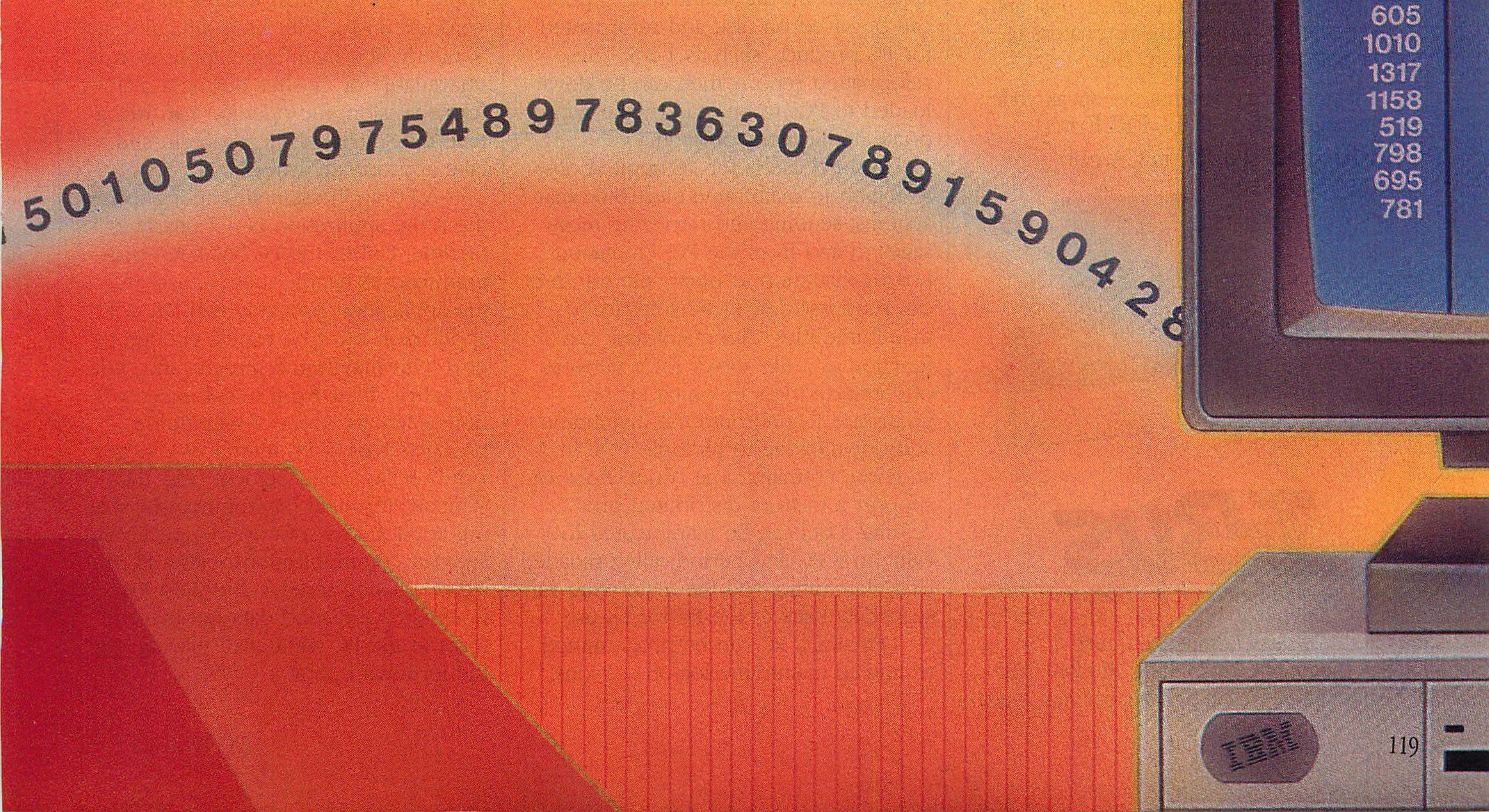
PAUL FIRGENS

Many office PCs are no longer strictly stand-alone machines. Increasingly they are being tied into corporate mainframe networks and expected to operate like 3270 terminals (with the help of emulator boards and software). Digital Communications Associates, Inc. (DCA) was a pioneer in this market with its IRMA board, which provided the PC with a coaxial connection to a network, terminal emulation, and file-transfer capabilities.

But IRMA is not enough for many users. Because a PC is much smarter than the average terminal, it was inevitable that users would want to take advantage of the PC's flexibility in connection with the mainframe's vast data and networking capabilities. This can be done, to a limited extent, but it requires some effort and involves compromises. Most of the compromises result from the fact that the mainframe and its applications are designed to communicate with a terminal, not another computer. Much of the effort results from attempts to control the terminal emulator. Until recently, custom routines had to be written to manipulate screen buffers and I/O ports, but a new software product from CDI Systems, Inc., AutoKey/3270, eliminates some of that work.

Sophisticated PC-mainframe applications attempt to satisfy one or more of four goals. First, they try to present a friendly interface for the user. A mainframe application usually requires many keyboard entries just to log-on to an application. For example, to use an application running under IBM's Customer Information Control System (CICS), which is a terminal, file, and program handling monitor, a user typically must invoke the monitor itself through the network, log-on to the monitor, then start the application, which may require another log-on. Each step requires a password, which often differs from the preceding password. In addition, the screens frequently are cryptic. For example, CICS log-on and log-off procedures present a blank screen onto which the user must

ILLUSTRATION • ANDY LEVINE



enter a transaction identifier—with no prompt displayed.

Routines can be designed to hide most of these steps, so that the user need only start the application while the PC and a mainframe program manage the overhead. These routines can perform the repetitious key entries, bringing the user directly to the initial application screen. A PC-mainframe application can hide the mainframe environment from a user by basing the interface entirely on PC menus. The application also can handle communications with the mainframe, filtering the communication to present selective information to the user. Mainframe software manufacturers often use this technique to allow PC-based products to work with their mainframe products.

Many manufacturers control both ends of the connection by creating special mainframe routines to communicate with their PC products. By using such routines, they are not forced to depend on standard text-based screen displays to transfer data, and can use compression techniques to speed data transfers and other communications between the computers.

The second goal of PC-mainframe applications is to transfer files for use by PC programs. This would allow, for example, Lotus 1-2-3 users to incorporate data directly from mainframe databases into their PC spreadsheets. It is not uncommon for financial managers to key data into spreadsheets from mainframe hard-copy reports—a waste of a company's resources and an extremely error-prone process. Using an application on the mainframe, reports can be saved on disk and reformatted for PC products such as 1-2-3. These reformatted reports then can be downloaded to PCs to be read by the appropriate applications program.

A third use is for PC applications to log-on to mainframe electronic mail systems, sending and retrieving messages to and from the PC. Emulation boards that support binary-file transfers can send backups of PC software to mainframe files. The mainframe can function as a file server for a network of connected PCs in which PC programs are loaded onto the mainframe and downloaded to multiple PCs. In addition, PC-mainframe connections can support a form of distributed processing. Data can be manipulated locally on a PC and periodically uploaded to a mainframe; and the mainframe can send data back to the PC for its use.

Finally, a PC can be programmed to initiate mainframe log-on sessions, execute transactions, and transfer data without user assistance. This capability permits long-running jobs and those that place heavy loads on the mainframe to be delayed until off hours when mainframe resources are more readily available.

A VIEW FROM THE MAINFRAME

The PC-mainframe environment is dominated by mainframe myopia—a mainframe cannot detect whether it is communicating with another computer or a human operator. The 3270 network was not designed to connect to computers; it was intended to transmit at most a couple of thousand bytes at a time to each of its terminal devices. Because the connection is display- rather than data-oriented, a PC application must view the mainframe world from a 24-by-80-character window. (See "Exploiting the 3270 Connection," Mary DeWolf, July 1987, p. 94.)

This orientation imposes restrictions and limitations. For example, the window usually has fields and headings appropriate for viewing and input from a terminal operator. When retrieving and sending data, an application must know the location of the appropriate screen fields. This requires some care and can be a major source of frustration when developing a PC application to link with the mainframe. It also makes error handling difficult.

A terminal operator usually has no problem adjusting to screen changes caused by software enhancements. If a field placement changes or extra menus are added to the mainframe application, a human user usually can work around it with little difficulty. Similarly, if a mainframe application is unavailable or crashes during a mainframe session, the operator can recognize the source of the problem and take appropriate action.

A PC application attempting to run the same application is not nearly as flexible. Unless the PC has been programmed to handle these situations, the application probably will fail. A PC application must be robust enough to deal with mainframe screens and situations that differ from initial expectations. Manufacturers of mainframe products having PC interfaces can provide a major enhancement over the standard terminal environment by writing applications software to specifically support communications with a PC. Instead of using screens and field locations, they can send data in binary form to the PC without displaying it to the terminal operator.

CARBON COPY PLUS. NOW PLAYING AT A FORTUNE 500 COMPANY NEAR YOU.

Our Carbon Copy™ communications software is a big hit with Corporate America. So big that it's already being seen in over 80% of the Fortune 500. That's why they're so eager to get their hands on Carbon Copy PLUS,™ the sequel without an equal.

Running on an IBM® or compatible PC, Carbon Copy PLUS includes enhanced remote control capabilities over and above those found in our original Carbon Copy. Plus terminal emulation, compatibility with Crosstalk® script files, and support of X-Modem and Kermit file transfer protocols.

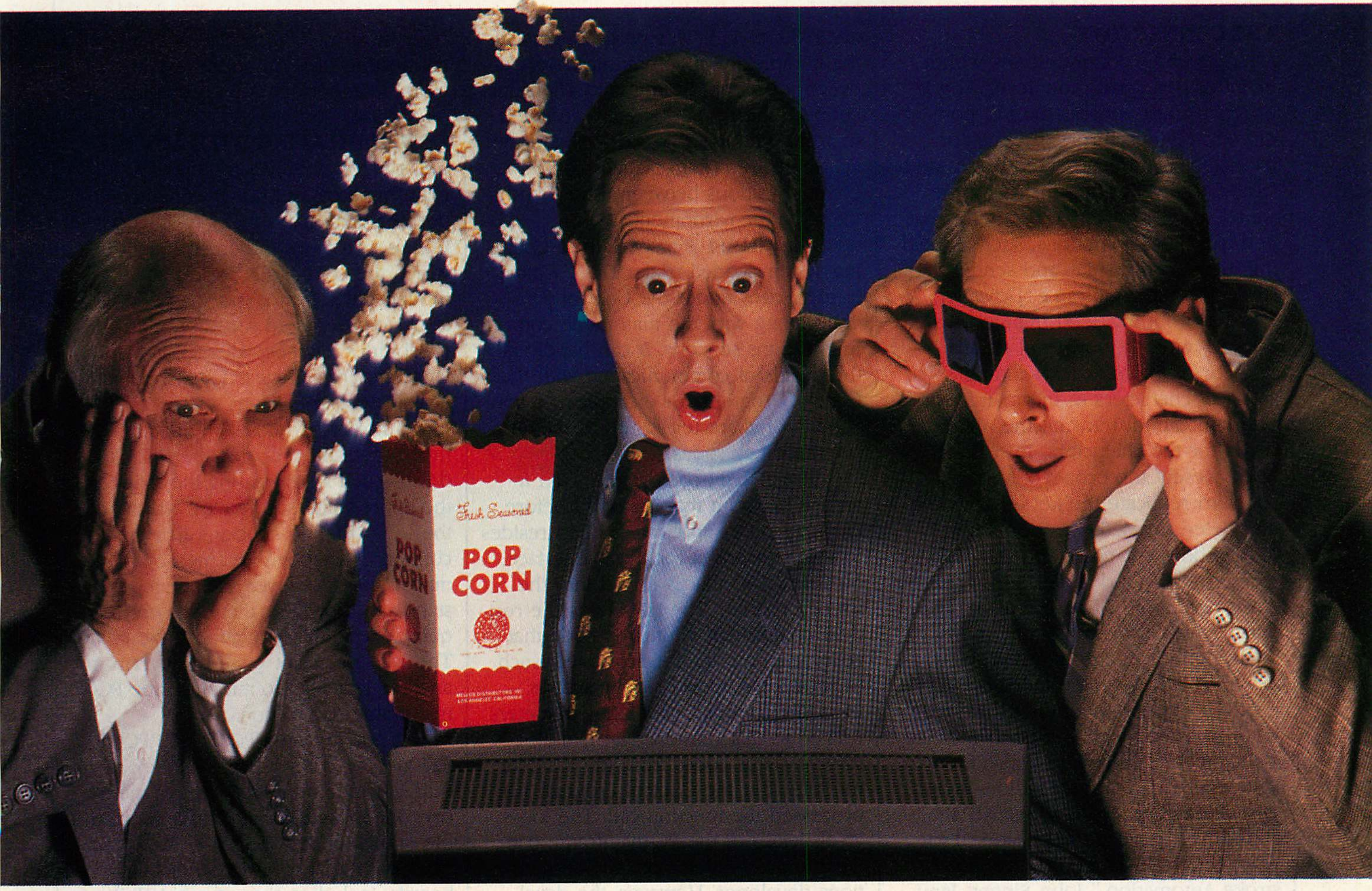
In remote control mode, use your PC to observe and control a remote PC as though the two were one. Applications include remote training and support. Access to your home or office PC while on the road. Remote access to a LAN. Remote access to a mainframe or mini via an IRMA® or SmartAlec® board. And much more. In terminal emulation mode, it's easy to access host computers and on-line information data bases. And you can receive or transmit telexes from around the world. Carbon Copy PLUS emulates most popular async terminals.

So, never again will you have to buy one communications program for PC remote control. And another for general-purpose communications. Because for about half the cost of buying both, Carbon Copy PLUS now gives you one integrated connectivity package that does it all.

For more information, contact Meridian Technology or see your local dealer. Carbon Copy PLUS. It's a program you definitely don't want to miss.

CARBON COPY PLUS.™

THE ONLY DOUBLE FEATURE IN COMMUNICATIONS SOFTWARE.

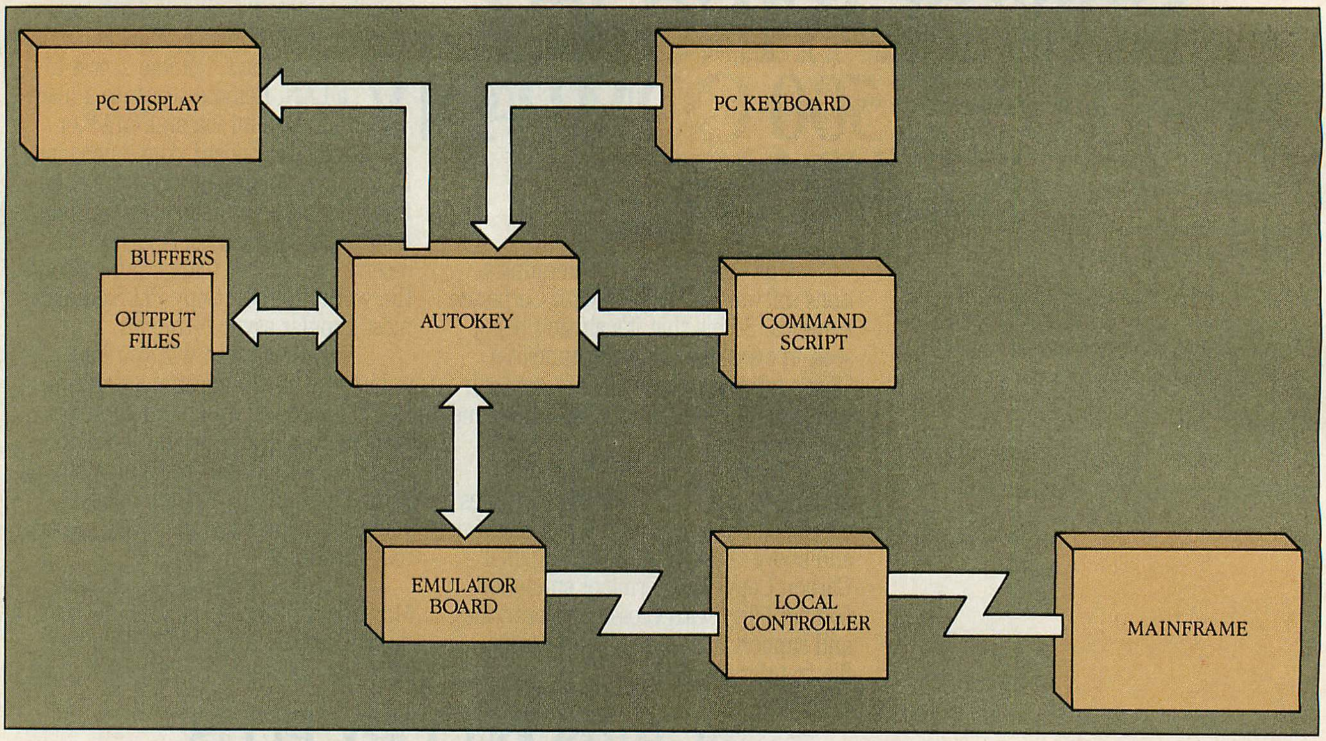


Carbon Copy and Carbon Copy PLUS are trademarks of Meridian Technology Inc. All other referenced products are trademarks of their respective manufacturers.

Meridian Technology
1101 Dove Street, Suite 120
Newport Beach, CA 92660 (714) 476-2224

CIRCLE 228 ON READER SERVICE CARD

FIGURE 1: *AutoKey/3270 PC-Mainframe Connection*



CD's AutoKey/3270 mediates the interaction between a user, a PC, and a 3270 terminal emulation session in a PC-mainframe environment. It simplifies the mainframe connection, offering a friendly interface for many complex operations.

THE CONNECTION

A PC generally communicates with a mainframe at a very low level. The PC program must address the I/O ports directly. Manufacturers of emulator packages have documented these interfaces fairly well, often including the source code for sample programs that perform basic operations, such as reading the screen display and sending keystrokes to the mainframe. However, they have not included high-level routines to mask the low-level details.

For example, when sending a string of characters to the mainframe, a program cannot transmit the ASCII codes for the characters. Instead, it must present the proper key-scan codes to replicate the keys that would have been pressed by an operator. These codes are unique to the 3270-type terminals and replicate neither the ASCII nor the EBCDIC character sets. Further, the codes must map more than one byte code per keystroke; for example, an uppercase letter is the code for the corresponding lowercase letter, prefaced with a code to indicate a shift-up key followed by a shift-down key code. This is rather tedious work.

One alternative is to use IBM's High-Level Language Application Program Interface (HLLAPI)—see "Keystroke Automation," John Singer, January 1986, p. 113. This program reduces the complexity of communications between a user application and a terminal connection. It was originally designed to work with the IBM 3270-PC and the PC Control Program, which handles communications between the PC and mainframe environments.

The HLLAPI is a specialized operating system loaded under DOS, which, in turn, runs under the Control Program along with the mainframe communications sessions. It can handle all the details of communicating with the Control Program, for example, the conversion of character strings to the appropriate key codes.

The HLLAPI has two parts: an interface program that is loaded before running a user application, then resides in memory, and a Language Interface Module (LIM) that is linked into the user's program. The LIM sends the resident component a function code that represents the service requested, such as **send keystrokes** or **copy strings from the terminal**, plus the parameters related to that request. The LIM also returns status codes from the resident module after the Control Program has processed the function call. The low-level details of the PC-mainframe connection are hidden from the applications developer. However, the developer must include the HLLAPI calls and connection in an application; the HLLAPI cannot operate on its own.

COMMUNICATING WITH AUTOKEY

CDI's AutoKey/3270 simplifies the PC-mainframe connection (see figure 1). It not only automates many tedious tasks, but also provides the environment in which the commands are executed. Although the software is not as flexible as custom-designed software, AutoKey's capabilities and specialized command language are sufficient for a wide variety of user applications.

AutoKey handles some details in much the same way that the HLLAPI does—such as translating the string to the appropriate scan codes, writing to the I/O port, and checking that the command has been accepted by the mainframe. Moreover, it operates as a stand-alone product that does not require a terminal emulation program. The user need not write a program in another language to automate the PC-mainframe connection.

It can run on a PC, PC/XT, PC/AT, 3270-PC, or compatibles. AutoKey requires a minimum of 256KB of memory, one diskette drive, and a compatible emulation board. It is designed to work with a large variety of such boards, including those from AST Re-

TABLE 1: *AutoKey/3270 Commands*

COMMAND / CMD ABBREVIATION arguments Optional parameters in square brackets: [. . .] Choices among options separated by vertical bars: l . . . l	**EXPLANATION**
ASGNCRSR / AC row column	Puts the cursor at the designated row and column on the 3270 display.
AT row col SAY "message string" [USING foregnd backgnd] AT row col GET "input string" [HIDE] AT row col DRAW BOX width BY height [SINGLE] l [DOUBLE] AT row col DRAW SPACES width BY height [USING foregnd backgnd]	Controls interaction with the PC. SAY places character strings on the display; GET prompts for input at the specified location and retrieves the input; and DRAW draws boxes and colors areas of the specifed size.
CALLFILE / CF "command file" ["subroutine name"]	Branches to the specified AutoKey command file.
COMPARE GET "character string" [UPPER] COMPARE BUFFER number "character string" [UPPER] COMPARE BUFFERS x AND y [UPPER]	Performs comparison between character strings stored in various locations. GET compares keyboard input to the specified string, BUFFER compares contents of a buffer (see READ) to the specified string, and BUFFERS compares contents of two different buffers. UPPER converts alphabetical characters to uppercase before the comparison is done.
ECHO [ON] l [OFF]	Echoes the AutoKey commands to the PC display as the commands are executed.
END	Terminates an AutoKey command file.
ENDRTN	Terminates an AutoKey subroutine.
FILE IS "filename" [APPEND]	Opens a DOS file and, optionally, writes at the end of the file if APPEND is specified.
GOTO label	Unconditional branch to the indicated label within the current command file.
GOTOFILE / GF "filename"	Unconditional branch to the indicated AutoKey command file.
HOTKEY	Switches between PC display and terminal emulation session.
IF "test" true-goto-branch false-goto-branch Test values: CURSOR l FOUND l STATUS l TIMER TIMEOUT l KERROR l EOF l MATCH	Tests one of the conditions above and branches to the specified location (either a label or NEXT to execute the command following the IF). The parameter tested is set by one of several AutoKey commands.
[KEY] "key value" [timeout value in seconds] [WAIT] l [NOWAIT] Key values to send: CLEAR l DOWN l ENTER l EOF HOME l LEFT l MARK l PA1-PA3 PF1-PF24 l PRINT l RESET RETURN l RIGHT l SYSREQ l TABBKWD TABFWD l UP	Issues a 3270-specific keystroke in the terminal emulation session.
KEYSTAT / KS	Checks and sets status of keyboard in the 3270 emulation session. Sets STATUS and KERROR test values (see IF).
PASSWORD / PS "character-string" "display message"	Pauses execution of current command file and then prompts the user for a password. If the entry matches the character string, AutoKey continues processing; if not, AutoKey aborts and displays an error message.
PAUSE "display message"	Pauses execution of the current command file, displays the specified character string, and waits until the user presses a key.
PRINT BUFFER "buffer number" [TO FILE [ENCRYPT]] l [ASCII] PRINT CHAR "ASCII decimal value" [TO FILE [ENCRYPT]] l [ASCII] PRINT #GET [TO FILE [ENCRYPT]] l [ASCII]	Moves data from a READ buffer to either a DOS file or to the PC display. It also can write an ASCII decimal character value, or the PC keyboard buffer contents from the last AT . . . #GET command to a file or the display.

search, CXI, DCA, Persyst, Forte, IBM, Quadram, and others. In addition, AutoKey can operate from remotely connected boards, as well as boards with direct coaxial connections. LAN versions of AutoKey have been released for some of the more popular networks, including Novell and Banyan.

CDI's AutoKey/3270 is available as either a developer's package or as a runtime system. The developer's package includes the command interpreter and utilities for installing the package and for encrypting and decrypting files, as well as a utility for determining the cursor position and the field attributes of a 3270 screen display. The AutoKey runtime system includes only the command interpreter.

Because AutoKey has no interactive mode, it requires a list of commands from which to work. These commands can be designed to control either the mainframe connection or the PC. When

COMMAND / CMD ABBREVIATION arguments Optional parameters in square brackets: [. . .] Choices among options separated by vertical bars: I . . . I	**EXPLANATION**
READ 3270 row column FOR length INTO BUFFER "buffer number" READ 3270 row column FIELD	Captures data from the 3270 display and places it into an internal buffer. A maximum of 1,920 characters can be captured and placed into any one of five available buffers. The command also can collect the contents of a display field, using the FIELD option.
RETRIEVE / RET row column ["text string for comparison"]	Tests to determine if the 3270 display cursor is at the specified location. Sets the test value CURSOR for testing with IF. An optional text string can be supplied and the command will set the CURSOR value to true if it finds that string in the given location.
SAVE CURSOR(n) R(n) C(n)	Saves the current 3270 display SAVE cursor location in an internal SAVE array, the cursor table. CURSOR saves both row and column location, R(n) and C(n) save only the row and column values respectively. These table entries can be referenced in various commands where a numeric literal would otherwise be required.
SEARCH "text string sought" [LAST] I [NEXT] I [NEXT n] SEARCH ALPHA I UNPROTECT I NUMERIC	Searches 3270 display for indicated string or field with the indicated characteristics.
TIMEOUT "number of seconds"	Sets the number of seconds that KEY will wait to be accepted by the mainframe before resuming command file execution.
TIMER START "number of seconds" TIMER CHECK	Sets and allows checking of a clock value against the current DOS clock time. Use of the CHECK parameter set the TIMER test value, which can be tested in an IF statement.
TYPE row column ["char string"] I [#record] I [#GET] I [#NEXT] TYPE * * TYPE R(N) C(N) [UPPER] I [LOWER]	Sends a character string to the mainframe at the location specified. The source of the string can be either the literal included in the command, a record from a TYPEFILE, the current string in the #GET buffer, or the next record from the TYPEFILE. The location of the input can be at the row and column specified in the command, the current cursor location (* *) or row and column values from the cursor table.
TYPEFILE / TF "filename"	Opens a DOS file for processing by subsequent TYPE commands. Records are read from this file by TYPE and appropriate subcommands.
WAIT UNTIL hh:mm [SHOW] WAIT UNTIL GET	Pauses execution of the command file until the time specified in the command is reached or until the time value entered though AT . . . #GET is reached. The SHOW option displays the current time and the time scheduled for resuming command execution.
KDOS ["filename"] I [#GET]	Suspends execution of the command file and starts a second copy of COMMAND.COM. Optionally, this command also can start a copy of a DOS program, which is specified on the command statement or through a previous AT . . . #GET command.

The script command language provides for the basic PC-mainframe manipulations most users will want to perform, as well as providing for other useful extensions. TYPE sends keystrokes to the mainframe. With TYPEFILE, it retrieves data from a DOS file and sends them to the mainframe. READ and PRINT retrieve and enter data from the mainframe.

AutoKey is loaded, it attempts to interpret and execute commands from either its default source, AUTOKEY.CMD, or from a file named as a command-line argument. These commands are stored in either an ASCII text file or in files that are encoded by AutoKey's encryption utility.

Depending upon the command chosen, AutoKey writes to the PC display and controls the host session. It displays messages to the user, receives and processes the user's responses, and uses colors and boxes to enliven the PC display. It also can initiate the terminal session and perform major emulator functions, such as sending keystrokes, retrieving data, searching for strings, checking the status of the keyboard, and positioning the cursor.

The command range is broad (see table 1). The scope of the language provides for the basic PC-mainframe manipulations, as well as other useful

extensions. The TYPE command is used to send keystrokes to the mainframe. With TYPEFILE, it retrieves records from a separate DOS file and sends them to the mainframe. Through READ and PRINT, it retrieves data from the mainframe session and puts the data into a file or onto the PC display. AutoKey also can encode this information as it is placed into a DOS file.

AutoKey's language includes both comparison and control-branching logic. User input can be tested from AutoKey/3270 scripts. Activity on the mainframe connection also can be tested from AutoKey/3270 scripts; for example, a script can test if the mainframe will accept keystrokes from the emulator as well as test for the presence of specific strings in the terminal display buffer. Subroutines can be called from a command file. These routines can be kept in separate DOS files to help structure and make modular the AutoKey command-file routines. An echo mode, available as a debugging aid, echoes the commands to the PC display as they are executed.

Other commands help make the presentation more useful and attractive. Some commands display messages and prompts; the AT command enables the user to draw boxes and borders in the colors offered by the IBM Color Graphics Adapter (CGA). Using HOTKEY, the terminal session can be presented to a user while the command session is suspended. A DOS program can be invoked from within a command-script execution. With WAIT, sessions can be suspended for up to 24 hours, and then run unattended.

AutoKey also provides several utilities. For security, ENCRYPT.EXE and DECRYPT.EXE encode and decode command files. AutoKey can interpret encrypted files directly, so that users need not decode their files before using them. Another utility, ADP.EXE, the attribute-display program, determines the row and column locations and attributes of the fields on a 3270 display screen. Without such a function, the developer of a PC-mainframe application often must determine this information by trial and error.

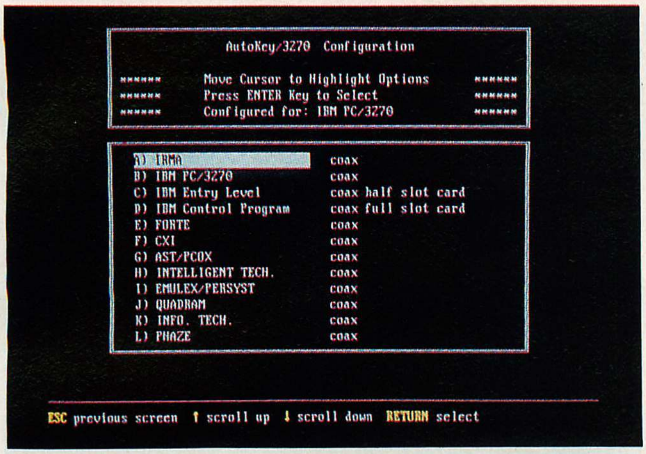
Before AutoKey can be used, it must be configured for the particular emulator board that is installed in the PC. A separate installation program leads the user through the process by a series of prompts. The first screen lists some of the boards supported by the package (see photo 1).

The process is straightforward if the user can answer the choices to be made. If the user is uncertain, AutoKey is not much help; the package provides little documentation about the configuration process. None is included online with the utility itself and the manual describes only certain boards in which the base I/O addresses can vary.

AUTOKEY TRIALS

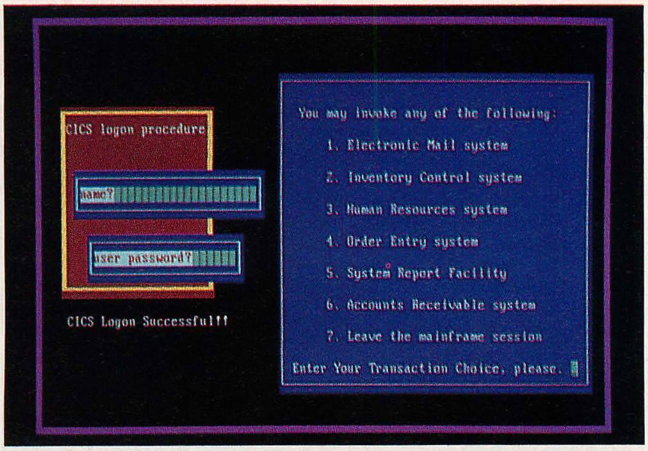
AutoKey version 1.2 was tested on a 3270-PC running DOS 3.2 and the 3270-PC Control Program 2.1, and on an XT with a DCA IRMA card running DOS 3.2. Although configuring the IRMA board was simple, the 3270-PC installation was not. The installation program is ambiguous about the appropriate choice for the 3270-PC. Experimentation eventually uncovered a setting that worked (sort of). The "IBM Control Program" option on the installation menu was the only one that worked with the 3270-PC. One of the major virtues of the 3270-PC is that it supports four simultaneous terminal sessions; however, it must run in distributed function terminal (DFT) mode in order to support these four sessions.

PHOTO 1: *AutoKey/3270 Configuration Screen*



The AutoKey/3270 configuration menu displays some of the emulation boards that are supported by the product.

PHOTO 2: *Sample Application Screen*



The display from an AutoKey program script demonstrates how it can control the display for a user.

AutoKey does not support DFT mode and gives no indication of this until it generates an error message at runtime and terminates. It turns out that AutoKey supports only the single-session, control unit terminal (CUT) mode. CDI plans to support the DFT mode in a future release of AutoKey.

AutoKey ran well for the most part. Nearly all of its basic operations, such as sending keystrokes, controlling the PC display, and performing logical tests, ran as expected. AutoKey supports calls to subroutines of script commands. The utilities performed as described. AutoKey also loaded and ran at an acceptable speed. Sheer execution speed in this environment is less important than in applications based solely on a PC. Here, the limiting factor is the time required by the mainframe to process the PC's requests.

A sample program is shown in figure 2, and the display generated by this script is shown in photo 2. In this program, the command file logs a user onto a CICS mainframe session, displays the user's transaction choices, and then waits for the user's selection. Once the user has chosen a transaction, the script invokes that selection and toggles to the mainframe session. The AutoKey's language is not complex; its

FIGURE 2: *Sample Command Script*

```
        cls
border magenta
at 6 4 draw spaces 22 by 12 using yellow red
        at 6 4 draw box 22 by 12 double using yellow red
        AT 7 5 SAY "CICS logon procedure" using white red
cicsloop:
        type "devlcics"
        key enter
        search "DEVLCICS SYSTEM"
        if found next cicsloop
        key clear
        type "CSSN"
        key enter
nameloop.
        search "NAME:"
        if found next nameloop
        at 10 6 draw box 27 by 3 using lcyan blue
        at 11 7 say "name?" using red white
        at 11 12 get 20
type #get
key tabfwd
at 14 8 draw box 22 by 3 using lcyan blue
        at 15 9 say "user password?" using red white
        at 15 23 get 6
        type #get
        key enter
at 19 5 say "CICS Logon Successful!!"

optloop:
at 4 35 draw spaces 44 by 20  using white blue
at 4 35 draw box 44 by 20 single using white blue
at 6 38 say "You may invoke any of the following:" using white blue
at 8 42 say "1. Electronic Mail system" using white blue
at 10 42 say "2. Inventory Control system" using white blue
at 12 42 say "3. Human Resources system" using white blue
at 14 42 say "4. Order Entry system" using white blue
at 16 42 say "5. System Report Facility" using white blue
at 18 42 say "6. Accounts Receivable system" using white blue
at 20 42 say "7. Leave the mainframe session" using white blue
at 22 37 say "Enter Your Transaction Choice, please." using white blue
at 22 76 get 1
chek1:
compare #get 1
if match next chek2
key clear
type "mail"
key enter
hotkey
goto optloop
chek2:
compare #get 2
if match next chek3
key clear
type "incx"
key enter
hotkey
goto optloop
chek3:
compare #get 3
if match next chek4
key clear
type "hrsx"
key enter
hotkey
goto optloop
chek4:
compare #get 4
if match next chek5
key clear
type "orex"
key enter
hotkey
goto optloop
chek5:
compare #get 5
if match next chek6
key clear
type "srfx"
key enter
hotkey
goto optloop
chek6:
compare #get 6
if match next chek7
key clear
type "acrx"
key enter
hotkey
goto optloop
chek7:
compare #get 7
if match next optloop
key clear
type "cssf logoff"
key enter
beep
pause
        cls
        end
```

This command file logs a user onto a CICS mainframe session, displays the user's transactions, and waits for the user's selection. It then invokes the selection and toggles to the mainframe. The language is not complex—anyone with even a passive understanding of programming languages should have little difficulty writing AutoKey command files.

command files should not be difficult for anyone with even a passive knowledge of programming languages.

MORE INFORMATION

One strike against this product is its documentation. For one thing, it should be more complete. The manual's descriptions are more like notes than a full-scale command reference. It briefly sketches the commands, with a few examples sprinkled in, but does not fully explain their use. The confusion that resulted from running under DFT and CUT modes is typical. Another example involves SEARCH—the documentation fails to mention that this command is case-sensitive.

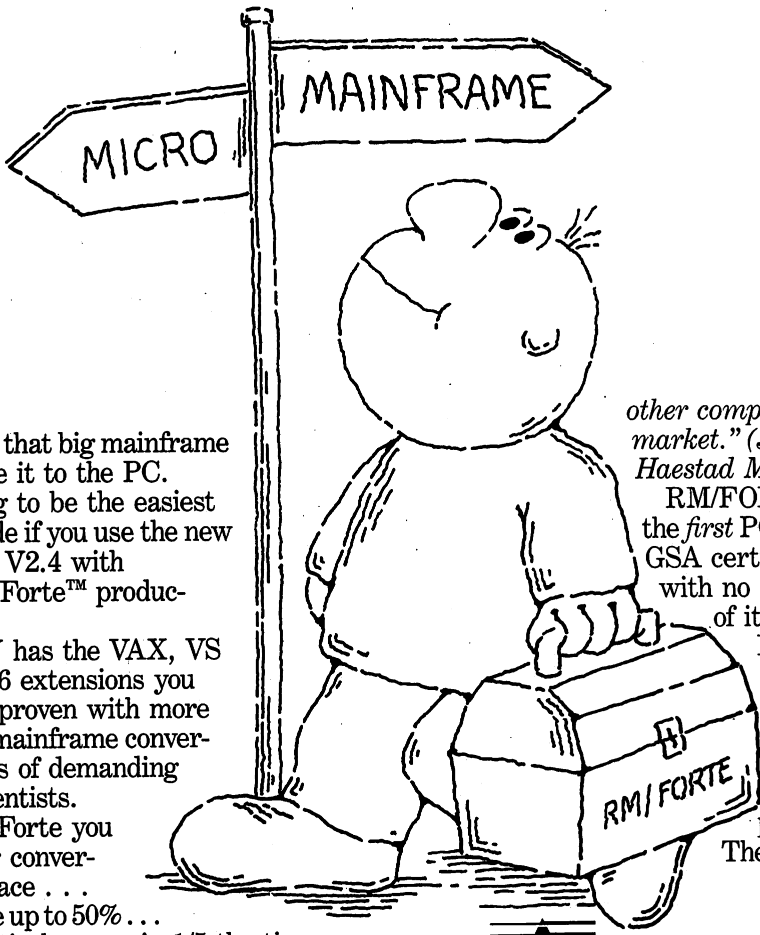
Many of the individual AutoKey commands depend on the successful completion of specific previous commands. For example, before the PRINT command can print to a file, that file must be opened with a FILE IS command. These dependencies are not made obvious in the documentation. Although these idiosyncrasies can be discovered fairly easily by experimenting with different scripts, users should not have to spend their time discovering and rediscovering unused and unavailable features because of omissions in the documentation.

The package does include a helpful on-line tutorial; one in the manual also would be useful. In addition, users have come to expect software to include executable examples that demonstrate the product. AutoKey has no such examples. It supplies a sample command file that is syntactically correct, but users should not expect this file to execute. Mainframe environments vary too much for generic examples to work in every situation.

SOFTWARE DIFFICULTIES

AutoKey can transfer data between a PC and a mainframe, but it is not recommended for moving large blocks of data. The READ 3270 command allows as many as 1,920 characters (a full screen of data) to be transferred at a time, but it would be awkward to move

FORTRANS-MIGRATION.



It's moving day.

Time to pack up that big mainframe program and move it to the PC.

Relax. It's going to be the easiest move you ever made if you use the new RM/FORTRAN™ V2.4 with Austec's new RM/Forte™ productivity tools.

RM/FORTRAN has the VAX, VS and FORTRAN-66 extensions you need, and is field-proven with more than two years of mainframe conversions by thousands of demanding engineers and scientists.

And, with RM/Forte you can complete your conversion at a record pace . . . cutting debug time up to 50% . . . diagnosing syntactical errors in 1/5 the time. You're in the fast lane because RM/Forte integrates all major program development tasks—edit, compile, link and debug—into one interactive, menu-driven environment. Just a few simple keystrokes move you from one task to another . . . and back again.

Plus, you get a full screen editor, a syntax checker and a source code manager that automates compiling and linking as you make changes.

RM/FORTRAN applications are more than just easy to develop—they run fast, too. Sieve or Whetstones, RM/FORTRAN benchmarks ahead of the bunch. And if you need it, RM/FORTRAN has a very fast 8087/80287 emulator to give you the power of a coprocessor.

But don't take our word for it. Listen to a moving professional: *"RM/FORTRAN lets us move mainframe programs to PCs faster and cleaner than any other compiler on the market." (John Haestad, Haestad Methods)*

OFFICIAL SUPPLIER IBM FORTRAN/2

RM/FORTRAN was the *first* PC FORTRAN GSA certified a Full Level ANSI 77 with no errors. You may have heard of its sister, IBM PC Professional FORTRAN.

Enough reading. Start writing. Fill out and send in the coupon below to find out more about RM/FORTRAN with the new RM/Forte productivity tools.

Then move it!

AUSTEC

Austec Inc.
609 Deep Valley Drive
Rolling Hills Estates, CA 90274
Or call 213/541-4828

PCT-9

Let's get moving. Send me details on RM/FORTRAN V2.4 with RM/Forte Productivity Tools, and your special upgrade offer.

Name ____________________

Company ____________ Title ____________

Street ____________________

City ____________ State ________ Zip ________

Phone ( ) ____________________

CIRCLE NO. 195 ON READER SERVICE CARD

large amounts of data in that way. The interpreter would have to pause the mainframe display with each screen's worth of data, write it to a file, then scroll forward for the next screen of data. A much more efficient method would be a file-transfer process, such as that supported by the 3270-PC. This process is not limited to working only within the bounds of the immediate screen—it basically lets the mainframe scroll the data on the terminal screen and lets the PC copy the display onto a file. (The 3270-PC utilities also speed the process by using file compression techniques; this also requires the assistance of a mainframe program.)

Another problem is that command files executing properly on one emulator board may not work correctly on another—some commands are not supported across all emulators. The SEARCH and HOTKEY commands perform reasonably well with the IRMA card, but neither works on the 3270-PC. CDI says that SEARCH does work on the 3270-PC, but that it takes a long time. This was attempted, but the program was terminated when the search had not completed after 30 minutes.

AutoKey's command language itself could stand improvement. Functions seem to be distributed arbitrarily among the commands, and some commands are overloaded with options. Why, for example, is only one command (AT) used to display messages to the PC display, retrieve input from the keyboard, and also paint colored boxes on the PC display? This can be confusing when trying to determine which command is appropriate.

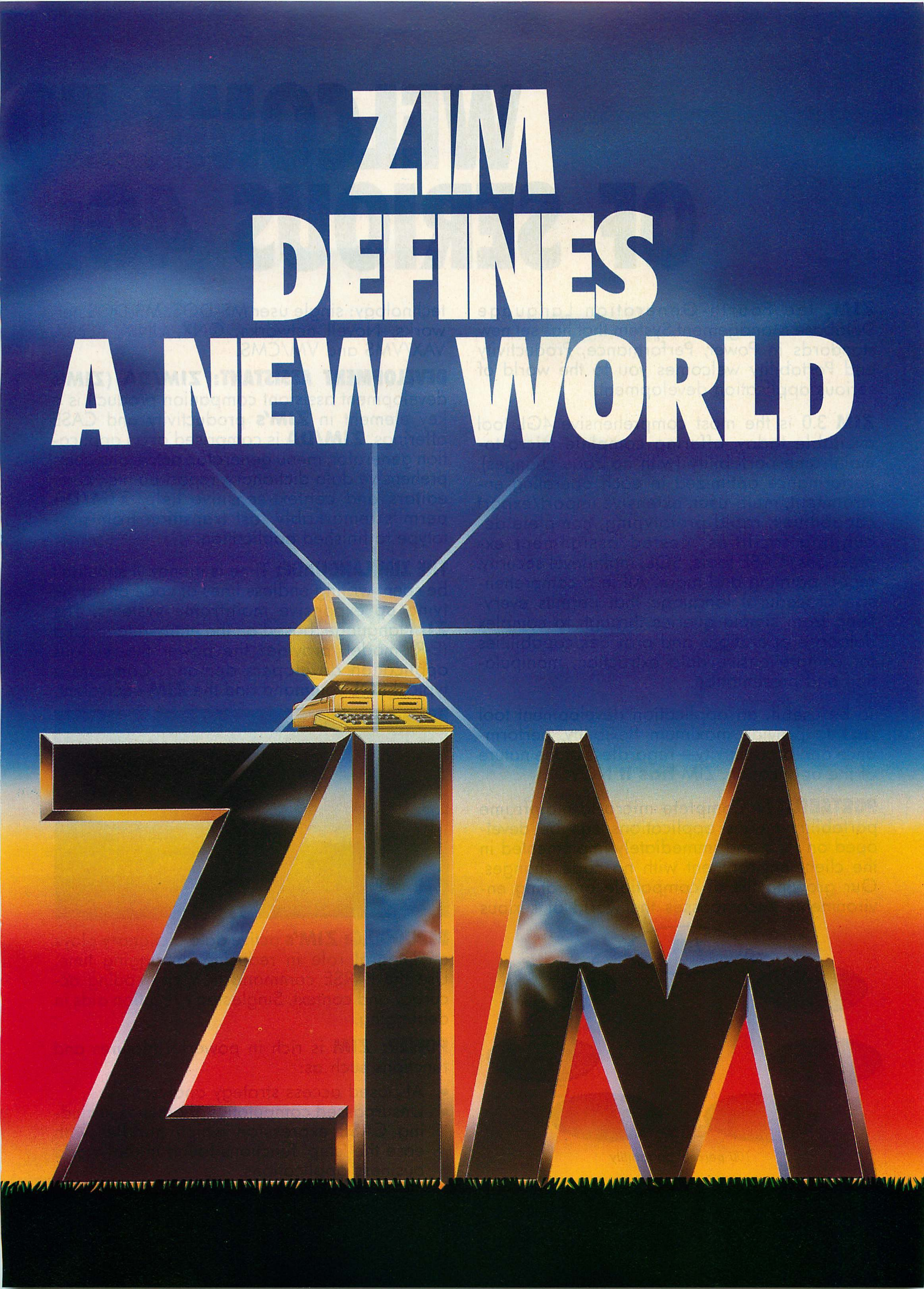
The language would be more usable with the addition of several features, such as parameter passing between calling and called routines. The language also cannot use variable names as arguments in the commands. It does, however, support references to the elements kept in a *cursor table*, an array of possible row- and column-screen locations, but the table is awkward to use.

But these criticisms may be a bit severe. AutoKey is not designed to be a complete programming language. Its intended environment, the PC-mainframe connection, is, itself, fairly fluid. Because mainframe screens and menu displays change frequently, it is important for a product such as AutoKey to allow changes to be made quickly, and AutoKey does do that. It can accommodate change more easily than the other automation methods because it is based on an interpreted script.

AutoKey is recommended for users who wish to begin automating a PC-mainframe connection. It hides the "native" communication modes required by emulator boards from the programmer and automates the connection at a fairly high level, saving users a great deal of work. AutoKey puts automation within the reach of users who are not inclined to attempt the low-level programming otherwise required. It also provides the means to develop a friendly user interface. Although the package is not flawless, its problems are insignificant compared to the overall usefulness it delivers.

CDI Systems, Inc.
2803 Butterfield Road
Oak Brook, IL 60521
312/574-5900
AutoKey/3270: Programmer, $149; Runtime version: $69
CIRCLE 353 ON READER SERVICE CARD

Paul Firgens, a senior database analyst for a Wisconsin firm, does extensive work in a PC-mainframe environment.


ZIM
DEFINES
A NEW WORLD
ZIM

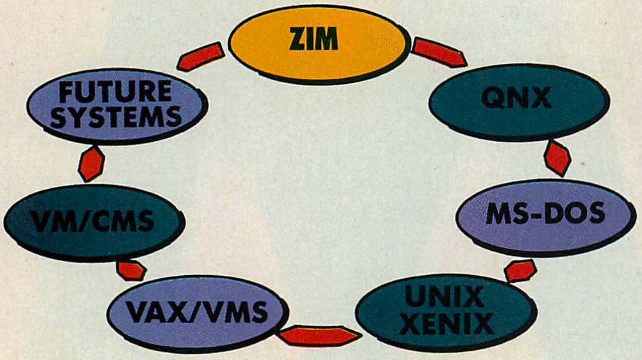
WELCOME TO
OF SERIOUS 4GL/

ZIM, the Fourth-Generation Language Database Management System that has set new standards in Power, Performance, Productivity and Portability welcomes you to the world of serious application development.

ZIM 3.0 is the most comprehensive 4GL tool available today, offering complete micro-to-mainframe portability (with no code changes), performance optimized in each operating environment, multi-user, extensive import/export capabilities, rapid prototyping, complete debugging facilities, nested assignment expressions, CASE tools, SQL, multi-level security, report painting and more. All in a comprehensive procedural language that permits everything from simple queries through to complex database operations and provides capabilities for comprehensive data extraction, manipulation and presentation.

To be useful, an application development tool **has** to provide maximum flexibility, performance and productivity—regardless of the nature of the application. ZIM **has it** NOW!

PORTABILITY: Complete micro-to-mainframe portability means applications can be developed on PC's and immediately implemented in the client environment with no code changes. Our growing list of compatible operating environments encompasses the world of serious technology: single user MS-DOS, MS-DOS networks, Novell networks, QNX, UNIX, XENIX, VAX/VMS and VM/CMS.



100 percent portability

DEVELOPMENT ASSISTANT: ZIM/DA (ZIM's development assistant companion product) is a key element in **ZIM's** productivity and CASE offerings. **ZIM/DA** is comprised of an application generator, menu generator, active and comprehensive data dictionary, report painter, code editors and context-sensitive help. **ZIM/DA** permits remarkably fast transition from prototype to finished application.

THE ZIM LANGUAGE: Time is money. It shouldn't be spent writing endless lines of code or worse, tying up expensive mainframe systems. The **ZIM** language, based on the entity-relationship (E-R) model, provides the power that serious application developers demand. Compare a typical SQL command and the **ZIM** equivalent:

SQL:
```
SELECT *
FROM WORKONTAB, PROJECTS, EMPLOYEES-
WHERE WORKONTAB.ENUM=EMPLOYEES.ENUM-
AND WORKONTAB.PNUM=PROJECTS.PNUM-
AND PROJNAME='ALPHA'
```

ZIM:
```
LIST ALL EMPLOYEES WORKON PROJECTS WHERE
PROJNAME='ALPHA'
```

DEBUGGING: ZIM's active data dictionary plays an integral role in reducing debugging time. Use the PARSE command to verify coding accuracy and context. Singlestep execution aids in debugging.

POWER: ZIM is rich in powerful features and functions such as:

- AI-based access strategy analyser.
- Unsurpassed computational abilities, including CASE expression which handles with ease the 'step' functions found in real-world business applications.

THE WORLD DBMS PROGRAMMING

- Natural multi-user features including user-defined transactions, roll-back, audit trails and roll-forward.
- Complete code-data independence.
- Exceptional text-handling capabilities.
- Unlimited forms manipulation and report-writing capabilities.
- Multi-lingual applications (French, German etc.)

SECURITY: ZIM's multi-level security features include log-in, passwords, permission-based access to entity sets, relationships and fields and encryption/decryption of entity sets and relationships.

VIRTUAL FIELDS: ZIM's unique virtual fields capability allows you to easily implement composite or concatenated keys, derived or computed fields, soundex indexes, etc.

IMPORT/EXPORT: ZIM masters the importing and exporting of screens and data to and from word-processing, spreadsheets and provides superlative forms composition through laser printers. **ZIM** applications already in use have saved millions of dollars through the reduction and/or elimination of paper forms.

ZIM is supported by an ever-growing family of companion products like **Niva's Guide to ZIM** (the authoritative guide from introductory to expert levels), the **ZIM Compiler** to ensure the highest level of performance of your **ZIM** application, a **Program Language Interface** permits "C" access to **ZIM** databases, **ZIM-ISQL** allows queries based on SQL syntax, **Runtime** and **Query Runtime** systems and **ZIMPLE,** an add-on module that permits easy use, allows end-users to create their own simple applications, ad hoc queries or paint their own reports (used with **ZIM** or **Query Runtime**).

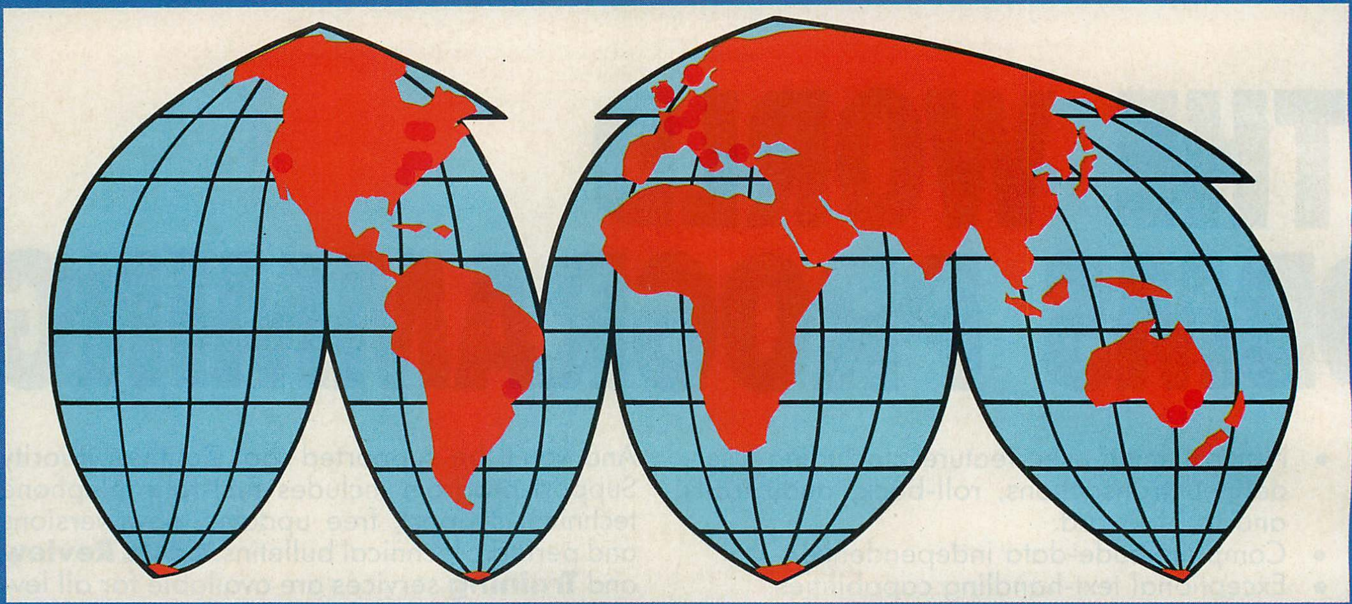
And you'll be supported too! Zanthe's Priority Support program includes first-rate telephone technical support, free updates, new versions and periodic technical bulletins. **Code Review** and **Training** services are available for all levels of **ZIM** users.

Around the world, from Australia to Europe, **ZIM Service Centers** and **Distributors** stand ready to supply and assist any **ZIM** user.

Already an integral part of data management in national governments, international banks, financial corporations and thousands of other applications **ZIM** welcomes you to the world of serious application development.

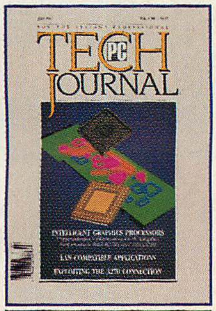
Zimpressive!

Zanthe Information, Inc.
1200-38 Antares Drive
Nepean, Ontario K2E 7V2
U.S. 800-267-9972
Canada 613-727-1397

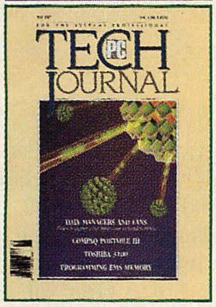


Complete your Library of PC Tech Journal...

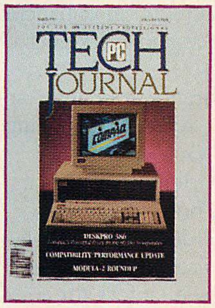
...Order those missing issues today!



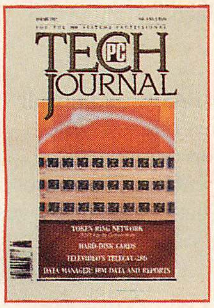
JULY 87
INTELLIGENT GRAPHICS PROCESSORS—Unprecedented sophistication for PC Graphics; First look at Intel 82786 and TMS 34010; LAN-Compatible Applications; Exploiting the 3270 Connection and more!



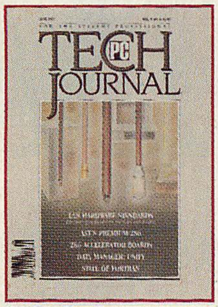
MAY 87
DATA MANAGERS AND LANS—Data integrity and multiuser considerations; COMPAQ Portable III; TOSHIBA 3100; Programming EMS Memory; Synchronizing Graphics Standards; Editors as Programming Tools; Command-Line Arguments for FORTRAN; Environment Variables; dBUG/EGA; Saywhat?; Diskit 2 Plus; Durapak; and more!



MARCH 87
DESKPRO 386—Compaq's Powerful Entry in the 80386 Sweepstakes; Compatibility Performance Update; Modula-2 Roundup; Multilevel Debugger; Binary Transfer; Procedural Algorithms in Prolog; Repertoire; Filemax; Mace Utilities; Bastoc; What Computers Cannot Do and more.

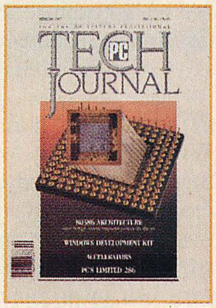


JANUARY 87
TOKEN RING NETWORK—IBM's Key to Connectivity; Hard-Disk Cards; Televideo's Telecat-286; Data Manager—IBM DATA and Reports; Mass-Storage Mergers; Display Adaptor Bottleneck; Realtime Systems: A Message-Passing Executive; Pixel Alignment of EGA Forms; Generic CADD; FANSI-CONSOLE; Above Disc and more.

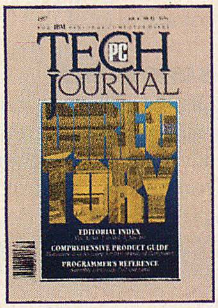


JUNE 87
LAN HARDWARE STANDARDS—Flexible Specifications for LAN Hardware; AST's Premium/286; 286 Accelerator Boards; Data Manager: UNIFY; State of FORTRAN: RGRAPH, automating CAD; Drawing Flexible characters; Batch File Interaction; MacII Attack; pcANYWHERE; ZVIEW; Command Plus; Applied AI and more!

APRIL 87
dBASE DIALECTS—Three Compilers support a de facto standard; ALR's Access 386; Multiuser Solution: CLASSIC; DOS Exception Handling; A CADD Workstation; A Resident Exception Manager for DOS interrupts 22H and 23H; DOS moves on the 80386; R-Card; Tom Rettig's Library; MEGA and more!



FEBRUARY 87
80386 ARCHITECTURE, Plus Windows Development Kit; Accelerators; PC's Ltd 286; ASYSTANT+; Speed Infusion; Token Ring Network Part II; Execution Profiles to the PC, Part II; Vector 87; ADC Model TD 440; High C; Database Practicum; Creating Sounds and more.



DIRECTORY ISSUE 86
(Volume 4, No. 13) Complete Editorial Index from Volume 1, No. 1 to Volume 4, No. 10; Comprehensive Product Guide for Hardware and Software for IBM Standard Personal Computers; Programmer's Reference Assembly Language Pullout Card; Company Index and more. Over 640 Pages!

DECEMBER 86—RT PC: Technology, Hardware and Software; IBM Cabling System; Basic Compilers; Large Turbo Pascal Arrays; Reading Locked Files; The JRAM Family; Programming for AI; Jump Searching; Locked Lists and more.

NOVEMBER 86—PROLOG ARRIVES Plus 3 Compilers with Power & Speed; LAN GATEWAYS; Circuit Design from P-CAD; Execution Profiles, The EGA Spectrum Part 2; UNIX Serial Devices; IBM Proprinter XL; Draflex 1; Determining Free Disk Space and more.

OCTOBER 86—EVALUATING THE EGA—Technology and Implementation; Hewlett-Packard Vectra; DOS at 5 years; Data Manager; TAS-PLUS; FORTH Programming; Visible RAM Disk; Dynamic Memory Management; Computer Law—The Virtues of Patience; EGA Paint and more.

SEPTEMBER 86—FORTH ARCHITECTURE AND INNERMOST WORKINGS; Interactive CAD; Clarion Review; Data Manager: DATABASE; 9 Multi-Port Boards Reviewed; Creating a Virtual Screen in Memory; Flicker Free Scrolling; Applied AI; Modula-2SDS; Modula 2/86 BLS; MIX C and more.

AUGUST 86—OUT FROM THE SHADOW OF IBM COMPATIBILITY AND PERFORMANCE ON 286 MACHINES; Compaq DeskPro 286; Nine Track Tape Systems; Animation with Sprites; Using ROM Characters; Sixteen Color Graphics; Quickcode for 1-2-3; Let's C and more.

JULY 86—ANIMATION TECHNIQUES—Optimal Performance on the CGA; 5251 Emulation; PASCAL Compilers; Data Manager; METAFILE; Video Function Call Fix; XTREE; The Nibbler; Macro Assembler 4.0; KSH-1; Scientific Calculator and more.

JUNE 86—SIX NEW SHAPES OF BASIC—Building on BASICA; Instant Screens; Data manager: Knowledgeman/2; iRMX for the PC; Mechanical CAD; RTDs and Thermocouples; Accessing the Print Queue; Whitesmith's C Compiler; Take Two; Zim Release 2.5; BASIC Development System and more.

MAY 86—DIGITIZING ANALOG DATA—Data Acquisition Boards; Breaking the 32 Megabyte Barrier; Data Manager: dBASE III Plus; VRTX; Finding Disk Parameters; The State of C Interpreters; Bit Rotation Speeds; Matching Regular Expressions; the Softstrip System and more.

QUANTITIES ARE LIMITED—ORDER TODAY!

Don't let there be a gap in your collection of expert coverage on applications, tutorials and product reviews from PC TECH JOURNAL. Limited quantities of back-issues are available. Please use the attached coupon, and enclose $7 per issue ($8 per issue for all foreign and Canadian orders, US currency only) and remit to:

K. Armstrong
PC Tech Journal
One Park Avenue—4th Floor
New York, NY 10016

Please send me the following issues of PC TECH JOURNAL. I am enclosing $7 per issue ($8 in Canada and all other foreign countries, US currency only).

☐ July 87	☐ April 87	☐ January 87	☐ December 86	☐ September 86	☐ June 86
☐ June 87	☐ March 87	☐ Directory Issue ($7.95 U.S./$9.95 Canada and Foreign)	☐ November 86	☐ August 86	☐ May 86
☐ May 87	☐ February 87		☐ October 86	☐ July 86	☐ Other ________

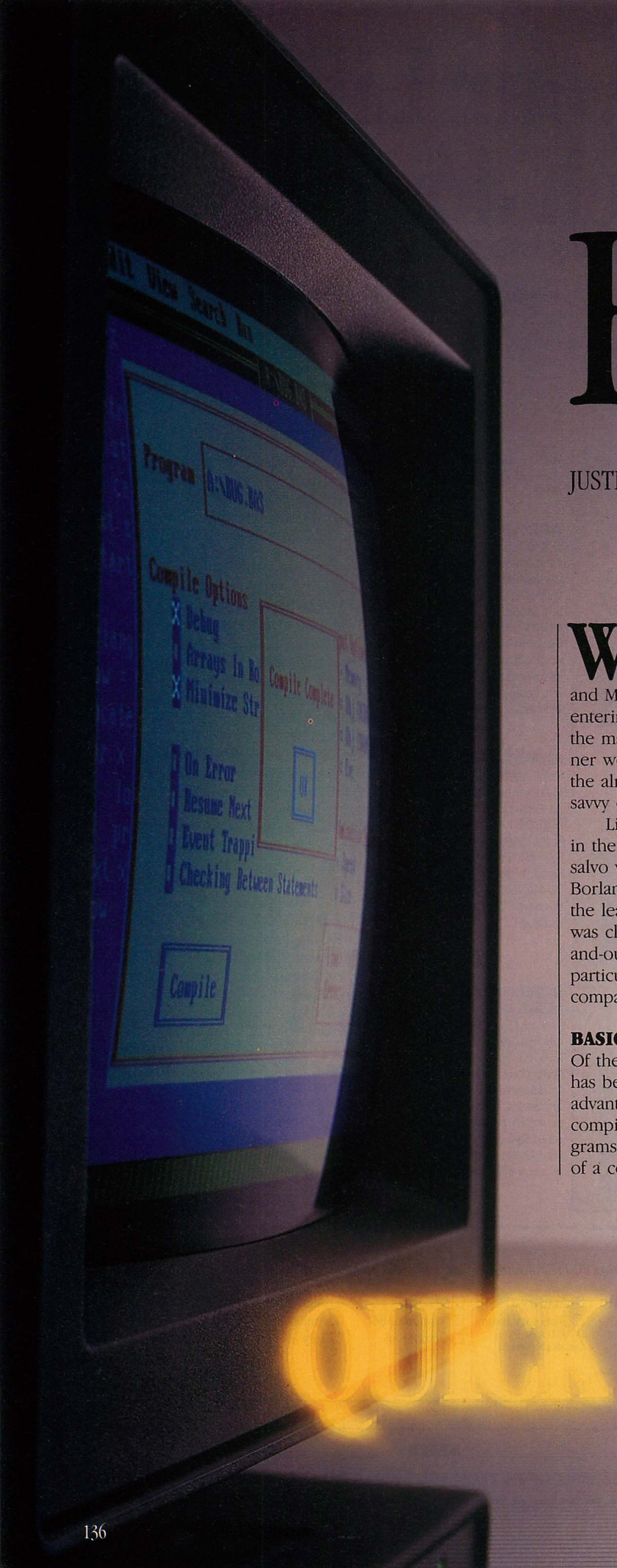
ENCLOSED IS $ ________

NAME ________

ADDRESS ________

CITY ________ **STATE** ________ **ZIP** ________

ORDER YOUR MISSING ISSUES OF PC TECH JOURNAL TODAY!



BASIC

JUSTIN CROM

When two major software vendors muscle up to the market with a new genre BASIC compiler, the products beg for a face-to-face comparison. Borland's Turbo BASIC and Microsoft's QuickBASIC offer integrated environments for entering, changing, compiling, and running programs—much in the manner pioneered by Borland's Turbo Pascal. A clear winner would capture the BASIC market, not to mention enhancing the already legendary programming brilliance and marketing savvy of the winning product's company.

Like any good spoiler in prefight preparation, our interest in the products during beta test period was keen. The opening salvo was fired by Microsoft in the form of QuickBASIC 2.0. Borland responded with Turbo BASIC and briefly looked like the leader. Once QuickBASIC 3.0 emerged, however, a draw was clearly in the making. In fact, neither is declared the out-and-out winner of the face-off, but each has strengths and, in particular environments, is recommended (see table 1 for a comparison of the features of the two products).

BASIC BASICS

Of the major programming languages, only BASIC traditionally has been interpreted, not compiled, and thus offers a major advantage to programmers. Without the intervening steps of compilation and linking, programmers can enter and test programs much faster. Consider, on the other hand, the advantages of a compiled language over an interpreted one: execution

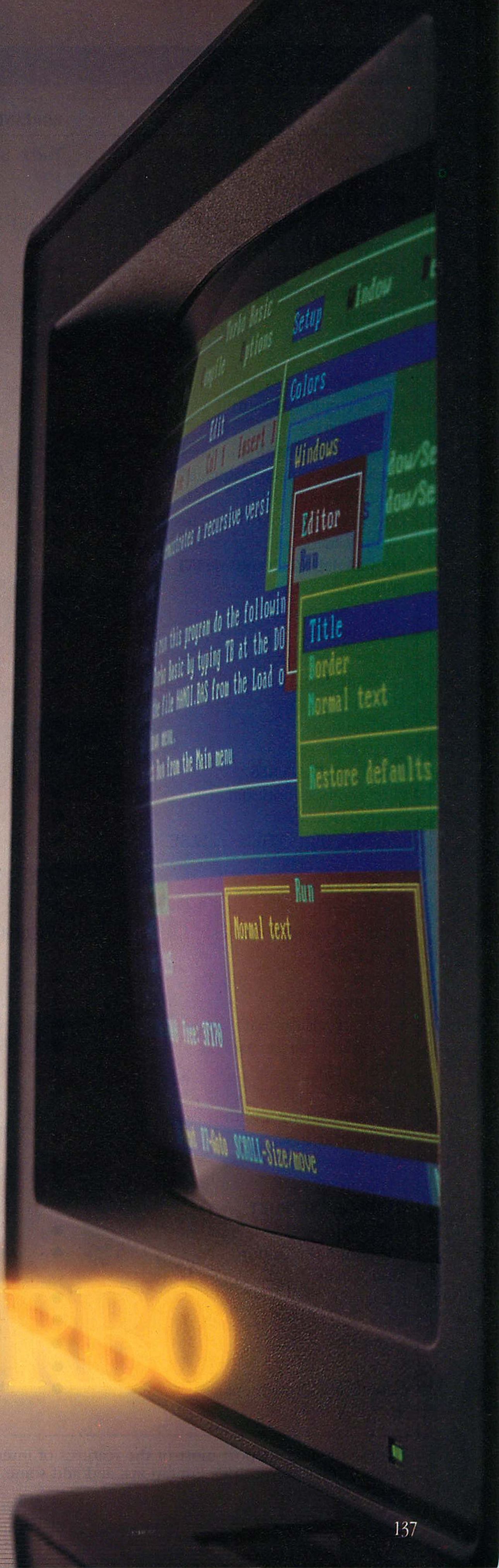
Face-off

Two software titans go head to head with a new genre of BASIC.

speed, security of source code, and no need for runtime support beyond that provided by an operating system.

The advantages of compilation have been available to BASIC programmers for some time and are thought to be improved enough to interest professional programmers. Five such BASIC compilers have been reviewed in these pages (see "Reconsidering BASIC," Marty Franz, December 1986, p. 142). However, those advantages usually are obtained by giving up the immediacy, quick turnaround, and ease of use of an interpreter. That is now changing. Turbo BASIC and QuickBASIC version 3.0 offer all of the advantages of compilation and, at the same time, give up very few of the amenities of interpretation.

Borland clearly has modeled Turbo BASIC after its very successful Turbo Pascal, even improving on that model when possible. The editor, compiler, and runtime system form a seamless, integrated environment that is at least as nimble and responsive as the traditional BASIC interpreter. With its windowing facilities and point-and-shoot menus, this is an up-to-date, Turbo Prolog-like implementation—a great improvement over the somewhat archaic interface of Turbo Pascal. Turbo BASIC offers support for math coprocessors, program and data spaces each exceeding 64KB, and creation of stand-alone executable files without the need for a linker. Like Turbo Pascal, this is the fastest, easiest-to-use implementation of its language. However, just as Turbo Pascal has limitations, such as not allowing separate compilation, so does Turbo BASIC.



PHOTOGRAPHY • WALTER LARRIMORE/BLAKESLEE-LANE

TABLE 1: *Compiler Features*

	BORLAND	MICROSOFT
Product	Turbo BASIC 1.00c	QuickBASIC 3.0
Price	$99	$99
BASIC FEATURES		
Required memory	256KB	320KB
Copy protection	○	○
DOS 2.x, 3.x support	●	●
Separate compilation	○	●
Output:		
.OBJ	○	●
.EXE	●	●
Needs linking	○	Optional
Needs runtime system	○	Optional
User libraries	○	●
Conversion of MBF numbers	●	●
Use with assembly language	●	●
Programs over 64KB	●	●
Data more than 64KB (see text)	●	●
Distinct string and numeric data spaces	●	○
PROGRAMMING ENVIRONMENT		
Full screen editor	●	●
Block commands	●	●
Find, replace	●	●
Case sensitive	●	●
Whole word	●	●
Mouse support	○	●
On-line help	●	●
Context-sensitive	●	○
LANGUAGE FEATURES		
Line numbers optional	●	●
Named labels	●	●
Multiline IF . . . THEN . . . ELSE	●	●
Loop constructs	3	2
Select case	●	●
Recursion	●	○
Length of variable names	No limit	40
DATA TYPES		
Integer	●	●
Long integer	●	○
Single precision real	●	●
Double precision real	●	●
IEEE format	●	●
String	●	●
Maximum string length	32,767	32,767
Dynamic arrays	●	●
Static arrays	●	●
Dynamic strings	●	●
HARDWARE SUPPORT		
8087 math	●	●
8087 emulator option	●	●
Mouse calls	○	●
Light pen	●	●
CGA	●	●
EGA	●	●
VGA	●	○
Sound (PLAY statement)	●	●
● = *Yes* ○ = *No*		

Both compilers support most of the features of interpreted BASICA, eliminate many of the interpreter's limitations, and add some features of their own.

Microsoft's QuickBASIC is the third version of this product. In many ways, it is a direct response to Borland's challenge in the area of compiled BASIC, but it goes beyond that challenge. The new Turbo-like features include support for math coprocessors and the implementation of many structured programming constructs. The beyond-Turbo features include a competent debugger and the ability to link object code from separately compiled modules that are written in QuickBASIC or assembly language.

With a reasonable price of $99, enhancements to the BASIC language, all of the advantages, and very few disadvantages of compilers, Turbo BASIC and QuickBASIC offer the user an escape from the limitations imposed by an interpreter. Both are vastly superior to the BASICA or GWBASIC usually obtained with DOS.

BASICA COMPATIBILITY

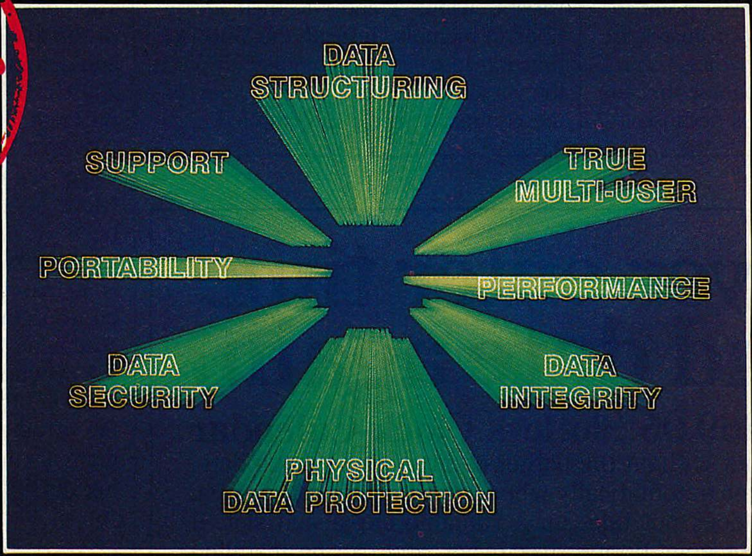
Of course, it goes without saying that the best features of a language are of little use if they are new and unfamiliar to the programmer. However, in this regard, Turbo BASIC and QuickBASIC are generally compatible with BASICA, and both fully support the interpreter's hardware control features. This includes peeking and poking anywhere in the 1MB address space, reading and writing I/O ports, and high-level statements for graphics and sound. Many BASICA programs can be compiled by either compiler without changes to the source code, but others require changes in order to allow for differences between the compilers and the interpreter. Some differences seem arbitrary. Turbo BASIC, for example, limits source code lines to 249 characters, whereas QuickBASIC and BASICA allow up to 255 characters.

Another difference is in the implementation of program chaining. Both compilers support only the plain CHAIN statement, without the DELETE, MERGE, or line-number options allowed by the interpreter. Therefore, when one compiled program chains to another, the target program always replaces the chaining program in memory, and execution always begins at the first statement in the target.

Other incompatibilities arise from the inherent differences between compilers and interpreters. For example, a compiler processes statements in sequential order, so any declarations such as DEFINT and DIM must physically precede the use of the declared variables. In an interpreter, declarations

MDBS III®

ATTN: APPLICATION DEVELOPERS



. . . DELIVERS THESE ESSENTIAL FEATURES. DOES YOUR DBMS?

MDBS III is more powerful than most mainframe data base management systems. . . and less expensive. MDBS III was designed for serious application developers like you. Like the developers of Solomon III, the "Number One" accounting system. And all the others who demand these essential features MDBS III provides:

DATA STRUCTURING—So flexible it captures any data relationship you can imagine. So comprehensive you'll design complex data bases faster than ever.

TRUE MULTI-USER—Few DBMSs give you as many facilities to guard against haphazard concurrent data modification as MDBS III does, down to the locking of individual data records.

PERFORMANCE—MDBS III gives you fast data modification and retrieval plus extensive performance tuning facilities.

DATA INTEGRITY—MDBS III provides airtight integrity assurances. . . from range checking to transaction-logging to enforcement of data relationships. . . *all automatically.*

PHYSICAL DATA PROTECTION—You get automatic recovery from media as well as from physical data destruction.

DATA SECURITY—Protect your data using passwords, encryption, and read/write access down to the field level.

PORTABILITY—MDBS III runs on a range of mini and micro computers, including LANs, and supports a variety of host language interfaces.

SUPPORT—**mdbs** is there when you need us, with in-depth seminars, telephone support, individual consulting and contract programming to help you develop and install your applications.

Call us today at 1-800-344-5832 for more information; in Canada or Indiana, dial 317-463-2581. Or write **mdbs,** P.O. Box 248, Lafayette, IN 47902. TELEX 209147 ISE UR.

MDBS III®

ABSOLUTE POWER

CIRCLE NO. 211 ON READER SERVICE CARD

mdbs®

must be executed before being used, but their placement in the source program is immaterial.

Another key difference involves the use of variables in subcommand strings for commands such as DRAW and PLAY. For example, the command

```
DRAW "M = X, = Y"
```

draws a line from the current point to the point whose screen coordinates are given by the values of variables *X* and *Y*. An interpreter can recognize the variables and find their values because it maintains a symbol table, or list of variable names and locations, at runtime. A compiled program does not have access to the symbol table because the compiler translates all variable names to addresses. A compiled program executes faster because it does not have to look variables up in a symbol table. At runtime, the program cannot relate the variable names *X* and *Y* to their locations in memory. Instead, the VARPTR$ function instructs the compiler to place the variable's address, not its name, in the command string. In compiled BASIC, the above example can be written in either of two ways:

```
DRAW "M = " + VARPTR$(X) + ", = "
  + VARPTR$(Y)
```

or

```
DRAW "M = " + STR(INT(X)) + ", = "
  + STR(INT(Y))
```

The compiler cannot recognize the use of variable names in command strings and automatically replace them with address references, because the variables need not appear directly in a DRAW or PLAY statement. The example DRAW statement can be replaced by DRAW M$, where the variable M$ has been previously assigned a string such as "M = X, = Y". At the time the string is assigned, the compiler has no way of knowing that it will be used in a DRAW statement. Furthermore, the variable M$ may not even have a value at compilation time; it may be read in from a file or otherwise constructed by a program at runtime. Therefore, the rewriting of DRAW and PLAY statements is usually the major effort of converting a program to run with the compiler.

Besides the issue of source-code compatibility, the issue of runtime compatibility must be addressed: will a compiled program produce the same results it did when interpreted? Expect two differences. First, because compiled code executes more quickly, any time-dependent behavior based on the execution time of interpreted code will be different. For example, many BASIC games run too quickly when compiled. Second, real numbers calculated with a math coprocessor (or a software emulation using the same IEEE numeric format) offer more precision than the Microsoft Binary Format (MBF) numbers used by the interpreter. As a result, numeric results can be slightly, or sometimes significantly, different.

In general, however, both source-level and runtime compatibility between the compilers and the interpreter are excellent. Effort spent to convert a program is more than repaid by subsequent performance.

LANGUAGE FEATURES

Both compilers provide significant extensions to the BASICA language. Some, like unnumbered lines, named labels, and structured programming constructs, are common to both, while others, like recursion and separate compilation, are offered only by one or the other. Programs taking advantage of the more advanced features of one compiler are not likely to be acceptable to the other;

of course, most programs written for either cannot be run with the interpreter. In the following descriptions, features not specifically mentioned as implemented in one or the other compiler are common to both.

Program structure. Line numbers are optional, and numbered and unnumbered lines can be mixed in the same program. Note that, unlike the line-number-oriented editor of BASICA, neither compiler's editor treats line numbers in any special way; any duplicate line numbers are not reported as errors until compilation, and lines are not sorted by number upon input. In addition, renumbering is not supported. Turbo BASIC supports the continuation of long statements over several lines, but the length must remain within the 248-character limit.

GOTO and GOSUB statements are supported, and their targets may be named labels as well as line numbers. However, many structured programming constructs are included that obviate the need for these "spaghetti coding" techniques. Conditional branching can be coded using either multiline IF . . . THEN . . . ELSE or SELECT CASE statements. In addition to the FOR . . . NEXT and WHILE . . . WEND loops of BASICA, DO WHILE and DO UNTIL loops are supported. The condition for controlling exit from the loop can be placed at the top or bottom of the loop; Turbo BASIC allows two conditions, one at each end of the loop.

The EXIT statement further reduces the necessity of using GOTOs. Turbo BASIC implements it whenever virtually any control structure is left prematurely: function, subroutine, IF, SELECT, or loop. QuickBASIC supports only EXIT FOR and EXIT DO.

Modular programming. To encourage modular structure, both compilers support two types of subprograms: procedures and functions. Unlike BASICA, functions can be defined over more than one line. Variables in a function, other than the parameters, are global by default, but can be declared local. In QuickBASIC, a function's definition must precede its invocation; Turbo BASIC does not impose this restriction.

Procedures are similar to the subroutines invoked by the GOSUB statement of BASICA, but like functions, they have names and can be invoked with argument lists. The implementation of procedures in the two compilers is very similar, but the scoping rules for variables are different. Turbo BASIC follows the traditions of the BASIC language; by default, all variables are global or known to all subprograms within the source file. Within a subprogram, however, variables can be made private by declaring them LOCAL or STATIC. Local variables are allocated at every entry to the subprogram and deallocated at exit, whereas static variables maintain their values between calls to the subprogram.

In QuickBASIC, variables are local to each procedure by default, but if explicitly declared as SHARED, their scope extends to every subprogram within one source file. Of the two scoping conventions, Turbo's is closer to the way BASIC has always been implemented and thus may be preferred by most BASIC programmers who have been using the language for some time. On the other hand, those with experience with other languages may feel that Microsoft's implementation finally makes BASIC a usable language.

In Turbo BASIC, both functions and procedures can be used recursively, which means that a subprogram can call itself. Recursion allows compact coding of many common numeric

and data processing problems, but it comes at the price of complexity and difficulty in following the program's logic. QuickBASIC does not support recursion, but (paving the way for future implementation) requires that all current procedures be declared STATIC or nonrecursive.

Because it supports separate compilation, QuickBASIC has stronger support for modular programming; subprograms can be compiled into separate object files and later linked into an executable program. Object files can also be collected into user libraries and retrieved automatically by the linker as needed. In Turbo BASIC, all subprograms of a module must be recompiled when a change is made to any one of them. This is made somewhat easier by the fact that subprograms need not physically reside in the calling main program file, but can be inserted into the main program file at compilation time with INCLUDE statements. QuickBASIC also supports INCLUDE.

Data types. Type declarations for numeric variables follow the BASICA conventions; they are optional, but can be made either by a DEF statement or by a suffix on the variable name. Turbo BASIC implements real numbers in IEEE format, QuickBASIC provides both IEEE and the MBF format used by BASICA and previous versions of QuickBASIC. The differences in format are significant primarily if a program reads file data prepared by, or writes data to a file for, a program that uses the other format. Otherwise, the only consideration is that IEEE provides more precision and therefore can produce different results in calculations. Both compilers provide functions for converting between formats.

For integer data, Turbo BASIC provides an extension in the form of a long integer type, a 4-byte number with a range from -2^{31} to $+2^{31}$ (a magnitude of more than 2 billion). Long integers are declared by the DEFLNG statement or by suffixing & (ampersand) to the variable name. This type can be used in applications requiring more accuracy than provided by real numbers, especially without a coprocessor. Accounting data kept in integer cents can represent sums of more than $21 million.

Both compilers offer named constants as an extension to the language. Turbo BASIC allows only integer named constants, which are declared by prefixing % to the name. Potential confusion exists in that integer variables have the same character as a suffix. In QuickBASIC, the CONST statement can name any constant, including the string type.

Strings are implemented in the standard BASIC fashion in these two packages. All strings are *dynamic*, meaning that their size does not need to be declared. The maximum length of a string is 32,767 characters.

Coprocessor support. By default, Turbo BASIC creates programs that automatically use a math coprocessor if one is present at runtime, or emulate it in software if not. A compile-time option is available to turn off generation of the emulation code for programs that will run only on machines equipped with coprocessors. Unfortunately, such programs crash hard when run on a system without a processor, requiring a cold boot to recover.

QuickBASIC offers more options. It is supplied in two versions: QB without coprocessor support and QB87 with. Programs compiled with the QB version use real numbers in MBF format, perform all calculations in software, and are forever oblivious to the pres-

ence of a coprocessor. The QB87 version of the compiler runs only on a system with a coprocessor; if started on a system without one, it produces an error message and exits gracefully.

Programs compiled with QB87 run with or without a math coprocessor, performing software emulation as necessary. Emulation is automatic for programs that use the QuickBASIC runtime module, but must be explicitly requested (by including an object module at link time) for programs linked into stand-alone modules. If a QuickBASIC program without the emulation module is started on a system without a coprocessor, it exits with a message, not a complete crash. Further QB87 options are available to force emulation even on machines with a coprocessor and to use MBF instead of IEEE format for real numbers.

Arrays. Both compilers limit individual arrays to 64KB, but allow as many numeric arrays as will fit in memory. Arrays can be either static (allocated at compile time) or dynamic (allocated when the DIM statement is executed at runtime). Dynamic arrays can be dimensioned with variable sizes or redimensioned during the course of execution. The QuickBASIC manual states that *large* numeric arrays must be dynamic, but does not define *large*. The limits on static arrays and on static data are discussed in memory usage section of this article.

Turbo BASIC supports arrays with up to 8 dimensions, while QuickBASIC theoretically supports 63. It is not possible, however, to have anywhere near that many and remain within the size limitation. The total number of elements in an array is given by the product of the number of elements on each dimension. For an integer array, the limit is 32,768 two-byte elements. With two elements per dimension (the minimum useful number), no more than 15 dimensions are allowed, simply because 2^{15} equals 32,768.

In QuickBASIC, as in BASICA, the index of the first element can be set with the OPTION BASE statement to either 0 or 1; the former is the default. Turbo BASIC allows setting the minimum subscript of all arrays to any integer value with the OPTION BASE statement and, as in Pascal, specifying both lower and upper subscript bounds for a particular array in the DIM statement. For example,

```
DIM Automobiles(1910:2000)
```

Both compilers (as well as BASICA) store array elements in column order. For example, the element following A(3,2) in memory is A(4,2). QuickBASIC also has the option of changing to row order so that the element following A(3,2) would be A(3,3).

Files. The compilers provide the standard set of BASIC file-handling statements for performing I/O on random and sequential files. Record-oriented I/O is performed as it is in BASICA, using FIELD statements for specifying the record layout. Neither language supports a compound data type for constructing records or structures.

To the standard capabilities, QuickBASIC adds file- and record-locking statements that permit shared file access in networked environments. Locks can be applied to entire files or to a range of consecutive specified records.

Turbo BASIC adds a binary file mode that treats a file as a sequence of bytes. In binary mode, a file pointer can be positioned at an arbitrary byte location within the file, and an arbitrary number of bytes can be read or written in one I/O operation. Although the equivalent capabilities can be pro-

grammed with standard random-access files, this method is both easier to program and more efficient.

Graphics and sound. Borland and Microsoft compilers both fully support the graphics and sound capabilities of BASICA version 3.2, with differences mandated by the nature of a compiler. The main difference is the need to replace variables in the command strings of DRAW and PLAY statements with VARPTR$ or STR$ functions.

All Color Graphics Adapter (CGA) and Enhanced Graphics Adapter (EGA) modes are supported, as well as screen-control statements for windowing and setting the orientation and logical dimensions of the on-screen coordinates. Turbo BASIC offers two Video Graphics Array (VGA) modes of IBM's PS/2 series: SCREEN 11 yields 640-by-480 graphics with 2 colors; SCREEN 12 is 640-by-480 with 16 colors.

Memory usage. Both compilers free the user from the BASICA memory restriction of 64KB for code and data together. Although they both advertise that machine memory is the only size limitation on programs and data, this applies only to the final program after linking together smaller components.

Turbo BASIC restricts to 64KB the source-code file that can be handled by the editor. Larger programs can be compiled by breaking them up into several files and then collecting them with $INCLUDE statements for compilation. If the code generated by the program exceeds 64KB, the programmer must manually break it up into segments of 64KB or fewer by inserting $SEGMENT statements into the code. Programs are limited to 16 segments. Although this is a reasonable limitation, it puts burden of tracking segment limits on the programmer and not on the program (where it belongs).

Hidden away in error messages (but confirmed by Microsoft technical support) is the fact that QuickBASIC also limits the code from one program module (main or subprogram) to 64KB. This limitation is rarely encountered because each subprogram is placed in its own segment. Segmentation is automatic as long as each program module is a reasonable length.

Each compiler manages data space differently. Turbo BASIC uses at least three segments, each up to 64KB. One holds the runtime stack and other data for the runtime system; the second holds the program's scalar data, string descriptors, and array descriptors; the third is for the program's strings (including string arrays). It also allocates each numeric array its own segment, each up to 64KB, and permits as many arrays as system memory allows.

QuickBASIC has one main data segment shared by the runtime system (for the stack and other overhead) and the program (for scalar data, string space, and static arrays). Only dynamic numeric arrays are allocated in distinct segments, as in Turbo BASIC. Thus, QuickBASIC has no distinct size limits for total string data and static arrays, because each depends on the other and on the total data requirements of the program.

Assembly language interface. The CALL statement in both compilers permits calls to assembly language subroutines, using the same syntax that is used for calling BASIC subroutines. In QuickBASIC, object code from assembly language procedures can be included at the link step.

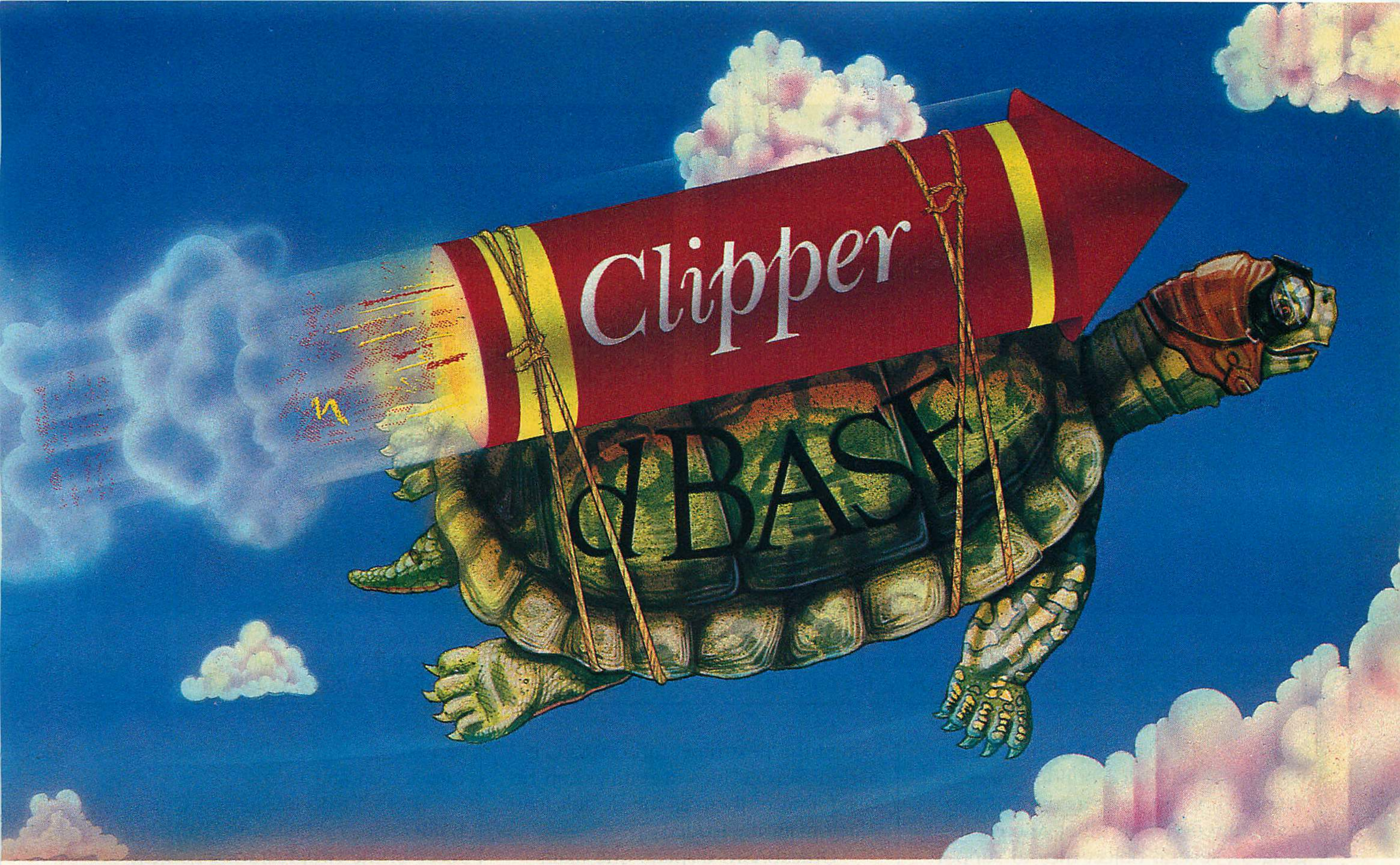
Turbo BASIC, however, has no link step, so calls must be resolved by routines within the same source file. An assembly language routine has the same header as a BASIC routine (a SUB statement), but its body consists of one or more $INLINE statements. As in Turbo Pascal, an $INLINE statement can contain constants that define bytes of machine code. As a useful extension, Turbo BASIC also allows the $INLINE statement to contain the name of a .COM file from which the code is loaded at compile time.

EASE OF INSTALLATION

Neither compiler is copy protected, and installation requires nothing more than using DOS commands to copy appropriate files to a diskette or hard disk. The instructions are clear and should pose no problems to users experienced enough to use a compiler. This installation method has two advantages: first, it holds no surprises with modified AUTOEXEC.BAT and CONFIG.SYS files that automated installation sometimes produces; second, the user can allocate files to directories of choice. Both compilers provide suggestions users are free to follow or not.

Turbo BASIC is provided on two diskettes, one of which contains sample programs. A working system consists of only the main TB.EXE file and a help file, which together total fewer than 250KB, so a diskette-only machine is quite practical for this compiler.

QuickBASIC's two versions, QB and QB87, each comes on two diskettes. A minimum installation consists of five files (the compiler, linker, runtime system, and two link-time librar-



Real programmers don't use dBASE. Or do they?

We're finding that some very swift programmers are using it to write some very fast applications, and are completing their projects much more quickly.

But they cheat.

They use our Clipper™ compiler to combine dBASE™ with C and assembler.

Turtle Souped

With dBASE used like pseudo-code, they can then quickly create prototypes that actually run.

Then, with dBASE doing the high-level database functions, they use our Clipper compiler to link in C or assembly language modules from their own bag of tricks.

And they're finding that they're linking in less than they expected because Clipper compiled code runs so fast and because of Clipper's built-in enhancements.

Clipper includes easy networking that provides file and record locking the way it should be done.

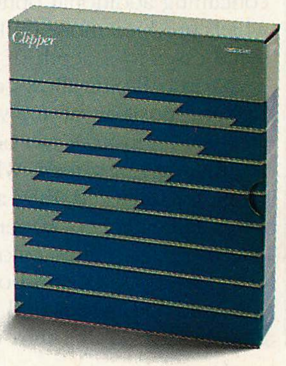
Fast screens that can be treated as memory variables and eliminate the need for direct screen writes and all that tortuous heap management code.

Box commands that make windowing a breeze. And more.

So if you'd like to use your time more productively, check us out:
Nantucket Corporation,
12555 W. Jefferson Boulevard,
Los Angeles, CA 90066.

Or if you're on deadline, call (213) 390-7923 today.

Clipper could get you out of the soup.



nantucket®

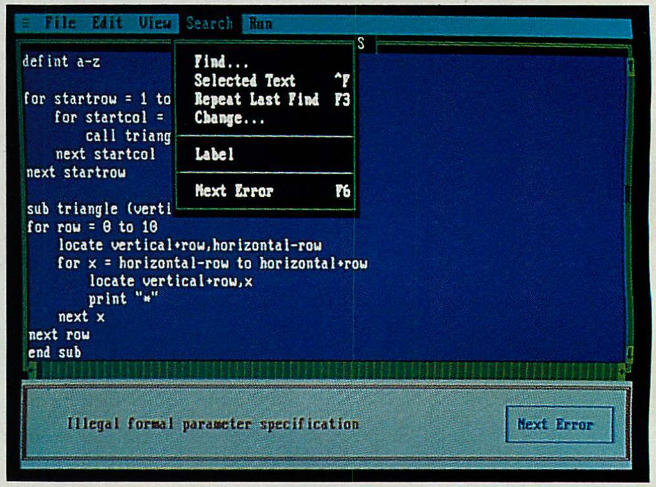
 In Europe: Nantucket Corporation (Europe) 2 Bluecoats Avenue, Fore Street, Hertford, Herts SG14 1PB Telephone 0992 554621.

CIRCLE NO. 224 ON READER SERVICE CARD

PHOTO 1: *Turbo BASIC Screen*

The default screen contains four tiled windows, the size, position, and colors of which can be changed. Pull-down menus appear when items are chosen from the top line.

PHOTO 2: *QuickBASIC Screen*



Windows are fewer and larger in QuickBASIC, with the window for current activity occupying most of the screen. Only the colors of the editing window can be changed.

ies) totaling about 500KB. In addition, each has a utility for building user libraries (not the same as the LIB program supplied with other Microsoft languages), several object files to support the assembly language interface, and some sample programs.

PROGRAMMING ENVIRONMENT

Superficially similar, the two compilers' programming environments reflect different designs. Flashy Turbo BASIC has colorful windows and a blizzard of pull-down menus. QuickBASIC is more restrained and even provides the minimalist option of invoking the compiler from the DOS command line (or a batch file) without entering the programming environment.

Turbo BASIC's initial screen has four windows for editing, compiler messages, program output, and trace output (see photo 1). The user makes selections from the main menu bar across the top of the screen by typing the first letter of the command or by moving a highlight with the arrow keys and pressing Enter. Upon selection of most main entrees, pull-down menus containing additional options appear, which produce more pull-downs—in some cases to a depth of four levels.

Turbo BASIC allows changes to the size and position of any window, and any or all windows can be zoomed to occupy the full screen. When all windows are full-sized, they are shuffled much like sheets of paper on a desk. The user can change color settings of text, backgrounds, and borders in both windows and menus and save the settings to disk so that they can be reloaded the next time Turbo BASIC is started. Several files of settings can be saved, one of them as the default that is automatically loaded.

QuickBASIC displays one main window at time, presenting a less cluttered appearance than Turbo BASIC (see photo 2). Its Macintosh-like interface is really designed for use with a mouse. Mouseless navigation through this environment can be tedious.

Users select from the main menu by pointing and clicking the mouse or by pressing Alt and the first letter of the function. Most selections produce a small pull-down menu, which may produce a large pop-up window for setting various options.

The advantage of using a mouse is especially evident on the screen that sets compile-time options. Without a mouse, selections are made by moving a highlight through the option fields with the Tab key; arriving at the desired option requires tabbing through all the preceding ones in sequence. In an effort to speed the process of mouseless selection, Microsoft has provided "shortcut" keys for some functions. Unfortunately, these key sequences can be difficult to remember, and once the menu has been pulled down to where the shortcut keystrokes are shown, they no longer work.

QuickBASIC provides for changing the colors of the edit screen, but not of the pull-down menus or option-setting screens. It does, however, allow the user to turn off color totally for use with a black-and-white monitor that is connected to a graphics board. With Turbo BASIC, colors would have to be turned off individually for each separate screen and menu.

Both Turbo BASIC and QuickBASIC offer on-line help. Turbo's is the nicer of the two, featuring context-sensitive help with a main directory and different pop-up screens for various help topics. The QuickBASIC help feature consists of a single screen summarizing the keyboard commands.

Editing. The Turbo BASIC editor uses commands similar to WordStar, Turbo Pascal, and many other text editors. This is a great feature for users familiar with that command structure, but combinations such as Ctrl-K-V for a block move or Ctrl-Q-A for search and replace functions seem rather arcane to those accustomed to other text editors. The key combinations are far from mnemonic, but they are ergonomically efficient, so once mastered the editing operations go quickly. Besides, an installation utility allows the user to redefine the command-key sequences.

QuickBASIC's editor, like the rest of that environment, is best used with a mouse. Scroll bars along the right and bottom margins of the window allow quick positioning: pointing halfway down the vertical scroll bar and clicking the mouse button positions the screen about halfway down the file. Marks within the scroll bar indicate the approximate position of the cursor relative to the start of the file.

Mouse-controlled block commands are especially easy; to define a block the user presses the mouse button at the start of the block, scans to the end, and then releases the button. A pull-down menu lists all block functions.

The process becomes more difficult and time-consuming if the block crosses screen boundaries, because the text must scroll until the end of the block enters the window. Search and replace commands are easily made by using the mouse cursor or cursor keys to select prompts on pull-down menus.

When running QuickBASIC without a mouse, the user selects text with a combination of Shift and arrow keys. A block that stretches over screen boundaries can be selected quickly with the Shift and PgUp and PgDn keys. Once the text is selected, the same edit menu used with the mouse is invoked by pressing Alt-E and moving through the menu with arrow keys. Even if a mouse is present, some operations can be performed more quickly with the key combinations.

The assignment of editor functions to shifted arrow keys could be a minor annoyance to users accustomed to using the Shift key to reverse temporarily the numeric state of the keypad. With NumLock off, pressing Shift-Up-Arrow does not type the digit 8, but instead selects a block of text.

In comparing the two editors, Turbo BASIC provides the added convenience of a status line that indicates not only the name of the file being edited but also the line and column numbers of the current cursor position. Without this information, the QuickBASIC user often flies blind.

Although both the Turbo BASIC and QuickBASIC editors are competent and offer a reasonably complete set of editing features, they lack some features of more advanced text editors. Neither editor can handle more than one file at a time, nor can it split the edit window so that different parts of the same file can be examined simultaneously. In addition, neither supports macros or wild-card search patterns. However, both editors produce straight ASCII text, so an external text editor or word processor can be used when such advanced features are required.

Filing. File-management options provided in both environments allow loading and saving files, printing source-code listings, and popping out to the shell to DOS. Once in DOS, nothing prevents the user from starting another copy of the compiler, but subsequent copies might be crippled by shortage of memory. Turbo BASIC has commands to change directories and to get directory information from within the compiler environment; QuickBASIC requires shelling to DOS to perform these functions.

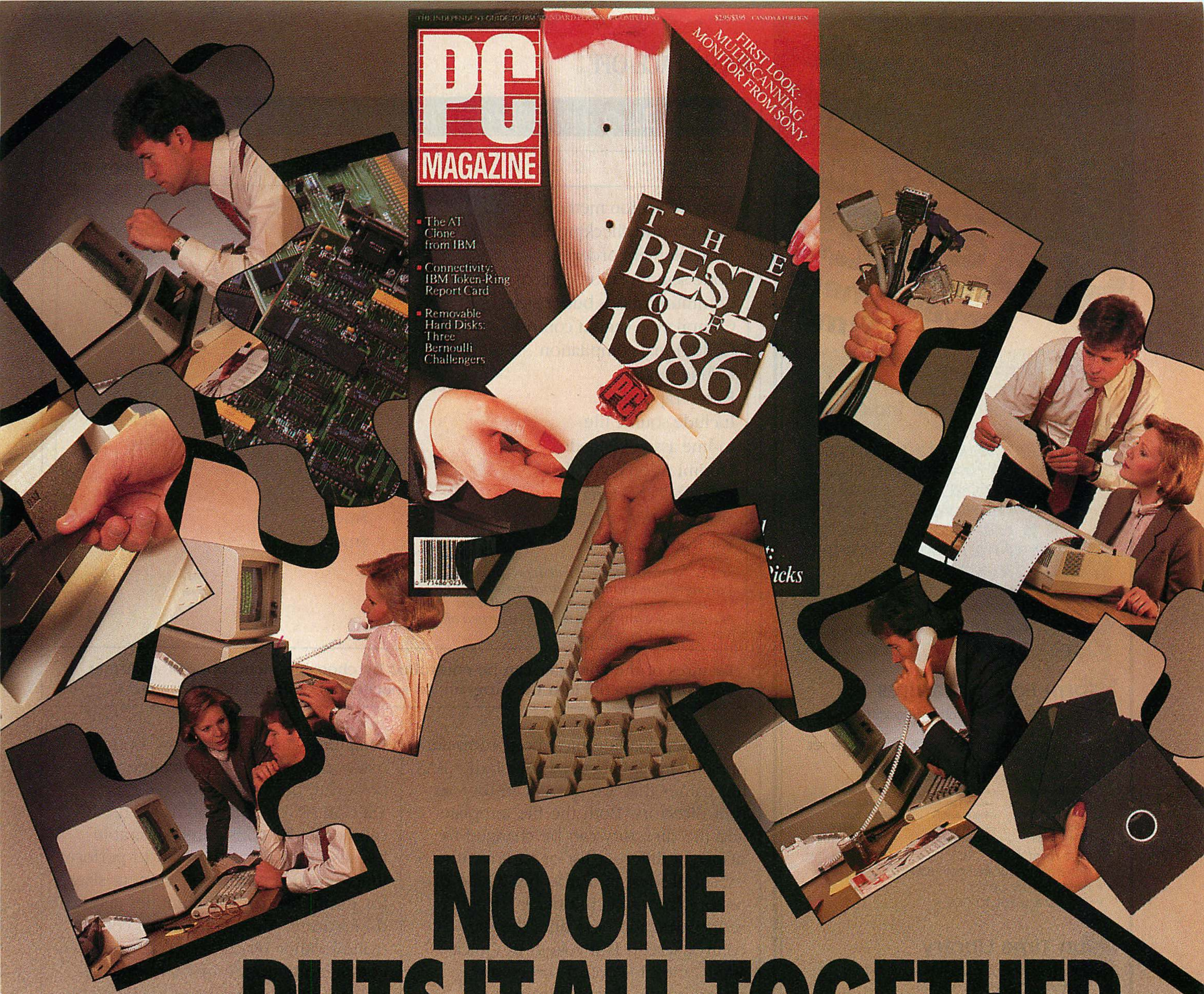

PC
MAGAZINE
FIRST LOOK:
MULTISCANNING
MONITOR FROM SONY
The AT Clone from IBM
Connectivity: IBM Token-Ring Report Card
Removable Hard Disks: Three Bernoulli Challengers
THE BEST OF 1986

TABLE 2: *Compilation Options*

	BORLAND	MICROSOFT
Array allocation method	$meta	$meta
Array bound checking	Menu	Always
Array element ordering	N/A	Menu
Code segmentation control	$meta	N/A
Communications buffer size	$meta	/param
Compilation output control	Menu	Menu
Conditional compilation	$meta	N/A
Debug mode	Menu	Menu
Event trapping	$meta	Menu
Include source file	$meta	$meta
In-line assembly code	$meta	N/A
Optimization control	N/A	Menu
Overflow checking	Menu	Always
Sound buffer size	$meta	N/A
Stack size	$meta	N/A

$meta = Metastatement within program
/param = Command-line parameter at startup
Menu = Set on menu prior to compilation
Always = Not an option; cannot be turned off
N/A = Not available

Features of the compiler can be controlled by choosing menu options, inserting control statements into the program, or specifying command-line parameters.

When loading files, both compilers present an alphabetized list of all *.BAS file names that are present in the current directory. Both the file template and the directory can be changed to display a different list or to limit it to a subset of files. In Turbo BASIC the directory from which a file is loaded is redisplayed the next time the file-control screen is invoked; in QuickBASIC the file-control screen always reverts to the default directory.

Once a list of files is displayed, typing the first letter of a file name moves the selection bar to the first file in the list starting with that letter. In QuickBASIC, repeated typing of the first letter advances the selection bar to the subsequent file names beginning with that letter. In Turbo BASIC, arrow keys are used for this purpose.

Turbo BASIC's file menu contains a SAVE option that can be dangerous: Turbo will, immediately and without verification, save the program from the editor under the name given when the program was loaded. To save a program under a new name without deleting the original requires a WRITE TO rather than a SAVE command. WRITE TO also warns that an existing file is about to be overwritten.

The SAVE command of QuickBASIC's file-control menu prompts with the name of the last file loaded and permits the name to be changed, but gives no warning if SAVE is about to overwrite an existing file.

Compiling. In general, more options make a compiler more flexible and powerful, but some, like the $SEGMENT directive of Turbo BASIC, can hardly be considered desirable because they force the user to perform functions (keep track of segment sizes, for example) that the compiler should do for itself. Most options are controlled either by choosing them from a compile-time menu or by placing directives (also called *metastatements*) in the code. Typically, the former is used for options that apply throughout a program; the latter for those that might apply to only a portion. The options that control the compilation process for both compilers are listed in table 2.

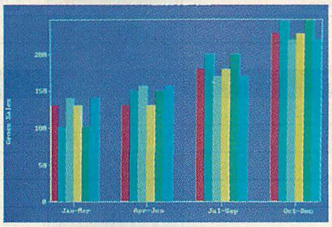
One interesting difference between the compilers is the way each requests the generation of event-trapping code to process ON KEY and ON ERROR statements. In Turbo BASIC, this is done by the $EVENT directive in the program. Not only can this localize such code (which exacts a penalty both in program size and execution speed) to the sections where it is needed, but more importantly it lets the program notify the compiler of the need for event trapping. In QuickBASIC, the user must specify this need by choosing from a menu; if he forgets, the program does not compile.

Because Turbo BASIC does not support separate compilation, all components of a program must be present at compile time. It does not need a

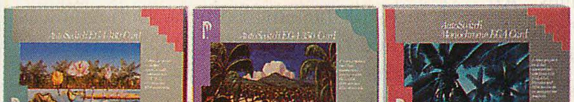
Next to IBM, Paradise makes more video controllers than anybody.

Quietly, without a lot of fanfare, Paradise has sold more than 1,000,000 PC video controllers.

Becoming the world's largest supplier of personal computer video cards and video controller chip sets outside of IBM.® In the process, we've built a record



of reliability and technical excellence that's second to none. We've introduced innovations like *AutoSwitching*™ and Automatic Monitor Detection.™ Pushed the resolution of IBM's graphic standards to new heights. Supported the most popular software applications. And created a product line-up that offers price and performance options designed to meet every



PC video need. From 480 line resolution EGA Cards to Basic Video Cards for your monochrome systems. That's why a lot of major computer manufacturers sell Paradise products with their systems. And why you should look for our name when you're ready for your next video card. We're Paradise, The Personal Computer Video Company. And we make more video controllers than anybody next to IBM. Call 415-871-4939 for the name of your nearest dealer.

PARADISE™

Paradise Systems, Inc. 217 East Grand Avenue South San Francisco, CA 94080 *AutoSwitch* is a trademark of Paradise Systems, Inc. IBM® is a registered trademark of International Business Machines Corporation. Copyright ©1987, Paradise Systems, Inc.

CIRCLE 230 ON READER SERVICE CARD

linker to resolve the references to various subprograms and can directly compile executable code, which can be placed either in memory for immediate execution or in a .EXE file on disk for subsequent execution from DOS.

QuickBASIC can also compile directly to executable code either in memory or in an .EXE file, provided that all subprograms are either in the same source file or available in a user object library. Alternatively, the output can be one of two kinds of object (.OBJ) file for subsequent linking, possibly with other objects from other compilations, into an .EXE file.

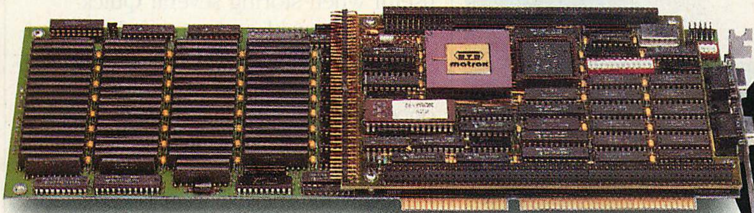
The ease of finding and correcting compile-time errors is a major advantage of an integrated compiler environment. In this regard, QuickBASIC has the advantage. The QuickBASIC compiler remembers up to 26 errors (the manual says 25) from each compilation. After an unsuccessful compilation, the editor positions the cursor at the first error detected, and a NEXT ERROR command is used to proceed to each one in sequence. This represents a substantial gain in efficiency over Turbo's one-error-at-a-time approach. When the Turbo BASIC compiler detects an error, it automatically invokes the editor and positions the cursor at the perceived cause of error. After each error is corrected, the program must be recompiled until it encounters the next error. Many errors require considerable time to correct.

The QuickBASIC compiler can be invoked noninteractively from the DOS prompt or from a batch file. Most of the options controlled by the compile-time menu can be set by command-line parameters. Error messages are displayed on the screen, but they can be redirected to the printer or a disk file.

Debugging. The debugging facilities of Turbo BASIC are hardly more advanced than those of BASICA, and seem at odds with the otherwise advanced features of the environment. Debugging is turned on either by a TRON statement in the source code or by choosing the TRACE ON option prior to compilation. The trace window displays the line numbers, labels, procedure, and function names as execution passes through them. Function keys allow toggling between continuous or single-step execution. Breakpoints cannot be set.

Debugging a graphics program in Turbo BASIC is all but impossible. Trace output jumps too readily from the confines of the trace window and writes all over the graphics screen, causing it scroll. The result is a chaotic

Fast. Faster. Fastest.



PG-1281 virtually redefines "high performance". With this new Matrox color display processor at the heart of an IBM PC AT, 386 PC, or compatible, a true professional workstation is born. Enhanced by four custom gate arrays, the PG-1281 is the fastest TMS34010-based board in the world today... and you have all the power of high-priced workstations at a fraction of the price!

▪ 1280 × 1024 resolution ▪ Full 64-bit bandwidth ▪ 65,000 vectors/second ▪ 100 million bits/second BITBLT ▪ Up to 1.5Mb on-board RAM ▪ Downloadable user code for easy application development ▪ Optimized drivers for major CAD packages ▪ Compatibility with all current graphics standards: Windows, X Windows, CGA, VDI, PGA, and more... ▪ Optional high-speed 3D coprocessor and EGA/VGA add-on modules.

If you need the best graphics performance available today, you need the PG-1281. For more details call us – fast!

1-800-361-4903

A LEADER IN VIDEO MICROTECHNOLOGY

matrox

In Canada, call (514) 685-2630. IBM and IBM AT are registered trademarks of International Business Machines Corporation. Windows is a trademark of Microsoft Corporation.

1281-PCT

jumble of painted areas, lines, and trace labels that defies deciphering.

When a runtime error occurs, Turbo BASIC reports the type of error and the value in the processor's program counter. If the program is running from memory, its source code is displayed in the editor and the cursor is placed on the statement causing the error. However, if an error is encountered in a stand-alone .EXE file, the source file is not available, and the editor is not in memory. In that case, the user can load the source file into the Turbo editor, turn on the runtime error option, and enter the program counter value from the error message. After the next compilation the cursor is placed at the point in the source code corresponding to the counter.

Compared to Turbo BASIC, the debugging provisions of QuickBASIC are excellent. When a program is executed with the debug option, a variable-sized window displays the source code with a highlight on the line about to be executed. The bar can then be stepped through the source code singly or continuously to provide a clear idea of the path taken during execution. The tracing either can be restricted to the main line of the program or can follow all paths through subroutines and functions. The window may obscure part of any graphics screen, but is always kept separate so the trace can be followed.

Because the stepping procedure is time consuming, QuickBASIC allows up to eight breakpoints in the source code to avoid having to step through error-free code. Execution can proceed at high speed until a breakpoint is encountered and the stepping procedure can be resumed.

The value of one variable can be watched while stepping through execution without including PRINT statements in the source code. A new watch variable can be named at a breakpoint, and it is displayed immediately when named, so although only one variable can be watched while executing, any number can be examined in turn while execution is suspended.

On an EGA system, QuickBASIC uses the 43-line mode to excellent advantage while debugging. For programs that operate in text mode, the top 24 lines can be set to display a full screen of output, with the remaining 19 lines used for scrolling the source code while debugging.

Linking. QuickBASIC allows separate compilation, an advantage for serious development work. The price for this flexibility is an extra link step to collect the separately compiled object modules into one executable file. The link process can be performed in one of two ways. The first collects into the executable file only the separately compiled modules, not any of the standard support routines from the library. The resulting .EXE file can be executed only in the presence of a runtime library called BRUN.EXE or BRUN8087.EXE with coprocessor support. The second method incorporates the necessary library routines into the program's .EXE file so that it can be executed without any runtime support. The type of .EXE file depends on the type of object files created at the compilation step.

Each type of .EXE file has its own advantages. The BRUN-module variety is significantly smaller and especially useful when storing several QuickBASIC executables on one disk. Instead of being replicated in every program, the support library is present only once. On the other hand, regardless of how many of its routines are actually needed, the whole support library is loaded when the program is executed, extending the load time and using up more memory. Also, if the BRUN file is not found in any directory on the path, the user must enter its location. Inexperienced end users should not be expected to keep track of files seemingly unrelated to their application.

The linker supplied with QuickBASIC is the one that comes with all Microsoft languages. It is functionally the same, although more recent than the LINK program supplied with all PC-DOS versions through 3.2. Linking can seem like a fairly complex process for the inexperienced, especially because it requires leaving the friendly confines of the QuickBASIC environment. Poor documentation, consisting of a scant one page of text without examples, does not help the situation. Only the command-line method of invoking the linker is given, and that method is easily prone to user errors. For example, placing an input file name where the linker expects an output file will overwrite the file. The manual does not mention that the linker can either prompt for files one by one or read a list of file names from a response file. More complete documentation on LINK is available in the DOS manual, except for DOS 3.3 where it was moved to the *Technical Reference*, which costs almost as much as QuickBASIC. Microsoft could vastly improve this phase of the development process by incorporating the link step into the compiler's environment.

COMPARING PERFORMANCE

Performance of the compilers was measured with a set of eight programs, four of which (SIEVE, MULDIV, HAT, and FILEIO) were used previously in a review of BASIC interpreters ("Six New Shapes of BASIC," Ted Mirecki, June 1986, p. 52). In order to provide better timing resolution at the higher speeds that can be achieved by compiled programs, the SIEVE and MULDIV programs were modified to increase the number of iterations.

Two new programs were devised to test the display speed in both text and graphics modes. SCRNTEST.BAS (see listing 1) uses the PRINT statement to fill the screen with text. The text is a string, so no time is spent converting numbers to characters. Correcting for the time taken to increment and test loop indexes did not materially affect the results. DRAWTEST.BAS (see listing 2) tests the speed of graphics using the DRAW statement. The design drawn is adapted from an example that appears in the IBM BASICA manual. Here, the overhead of the loops is subtracted from the drawing time. This test is a better indication of the graphics speed than the HAT program, because HAT spends more time calculating trigonometric functions than it does in the actual drawing process.

The SAVAGE.BAS benchmark (listing 3) tests execution speed and numeric accuracy of certain transcendental functions. It was adapted from the program published in *BYTE* (vol. 10, no. 11, 1986, p. 67). The last program, BIGTEST, was concocted specifically for testing compilation speed on large programs; it is described below.

All tests were run on an 8-MHz AT with 640KB of memory, an 80287 math coprocessor, and an IBM 30MB hard disk. Compiler options were set to produce maximum execution speed: debugging, event trapping, and error checking (array bounds, overflow) turned off and coprocessor emulation not included. The programs would not run on a system without a coprocessor.

The results are listed in table 3. In most of the compute-intensive tests (MULDIV, SIEVE, HAT), QuickBASIC is ahead by a small margin, but not enough to be noticeable without measurement. Turbo BASIC is 38-percent faster in the SAVAGE test, which borders on the significant. Turbo BASIC also has a slight but inconsequential advantage in graphics. The significant difference shows up in text display (the SCRNTEST program), where Turbo BASIC is faster by a factor of more than 10. Otherwise, no clear winner really emerges on execution speed.

Turbo BASIC produces .EXE files that are significantly smaller (by 20 to 35 percent) than the stand-alone files of QuickBASIC. However, QuickBASIC can produce files about 90-percent smaller for execution with a runtime system. Although an application consisting of one program can be smaller in Turbo BASIC than in QuickBASIC, the situation can be reversed when multiple programs are required. For example, the total size of the first seven Turbo programs listed in table 3 is 168KB, while the smaller QuickBASIC versions plus the runtime library (BRUN8087.EXE, 76KB) total 101KB.

These small benchmark files compiled so quickly that accurate measurements of compilation speed were impossible. Instead, a large program, BIGTEST.BAS, was constructed solely for this purpose:

```
A = 0
A = A + 1
(above line repeated 2,000 times)
PRINT A
```

The times for compiling this program to memory, to a .EXE file on disk, and (for QuickBASIC only) to a .OBJ file are listed in table 3. Turbo BASIC is clearly superior here, by a factor of two. For both compilers, the difference between compiling to memory and to a file are not significant. QuickBASIC takes slightly longer to produce a .OBJ file than a .EXE file, because object code is written to disk piecemeal throughout the compilation process, while the .EXE file is constructed in memory and written out all at once.

QuickBASIC's inability to compile a large file properly was one of the biggest problems encountered during testing. To test the limits of the compilers with large programs, the BIGTEST program was expanded by replicating the line **A = A + 1** several thousand times. With debugging options turned off, QuickBASIC generated the error message "expression too complex" for line 426, even though that line was exactly the same as the preceding 424 lines, which compiled correctly. With the debug option, QuickBASIC could compile a program of about 2,600 lines; larger files produced the message "critical error: compiler out of memory." Although nonsense, the program is syntactically correct and should compile without difficulty. Turbo BASIC handled up to 4,335 lines before generating a full 64KB segment of code, and it could generate even larger pro-

TABLE 3: *Compiler Performance*

	BORLAND	MICROSOFT
SIEVE		
.EXE file size: stand-alone	22.2	28.1
for runtime	N/A	3.6
Execution time per iteration	0.4	0.3
MULDIV		
.EXE file size: stand-alone	23.5	30.4
for runtime	N/A	3.8
Execution time, 50,000 iterations	23.0	20.1
HAT		
.EXE file size: stand-alone	25.4	38.7
for runtime	N/A	4.2
Execution time	181.6	107.3
FILEIO		
.EXE file size: stand-alone	24.1	32.8
for runtime	N/A	3.6
Execution time	119.1	118.9
SAVAGE		
.EXE file size: stand-alone	22.8	32.2
for runtime	N/A	3.6
Execution time, 25,000 iterations	29.5	36.1
SCRNTEST		
.EXE file size: stand-alone	21.8	31.1
for runtime	N/A	3.4
Execution time	0.8	9.2
DRAWTEST		
.EXE file size: stand-alone	28.6	43.0
for runtime	N/A	3.4
Execution time	15.3	17.4
BIGTEST		
Compilation time to memory	9.5	19.5
to .EXE file	10.3	20.1
to .OBJ file	N/A	26.0

N/A = Not applicable

QuickBASIC's slightly faster execution is barely noticeable in practice, while Turbo BASIC's compilation speed is definitely noticeable on longer programs.

grams if $SEGMENT statements were inserted in the source code.

Microsoft technical support had no answer to this problem, at first ascribing it to the 64KB limitation for code segments. The solution suggested was to replace the straight-line code sequence with a loop, but that did not address the reason for the failure.

The MULDIV test in QuickBASIC generated no error, while Turbo's error was an insignificant 10^{-16}. In the SAVAGE test, on the other hand, both Turbo BASIC and QuickBASIC produce the correct result of 25,000, even with single-precision numbers.

The SAVAGE program is especially good at showing the math coprocessor's effects on speed and accuracy. Running the test in software emulation mode (using IEEE format numbers) produced no error but took more than 8 minutes, compared with 36 seconds with the coprocessor. Using numbers in MBF format, QuickBASIC version 2.0 took 106 seconds and generated an error of 80 percent; interpreted BASICA ran 6½ minutes and managed to produce the identical error.

Turbo BASIC does have the annoyance of printing more digits than it has bits in representing the number. For example, the command

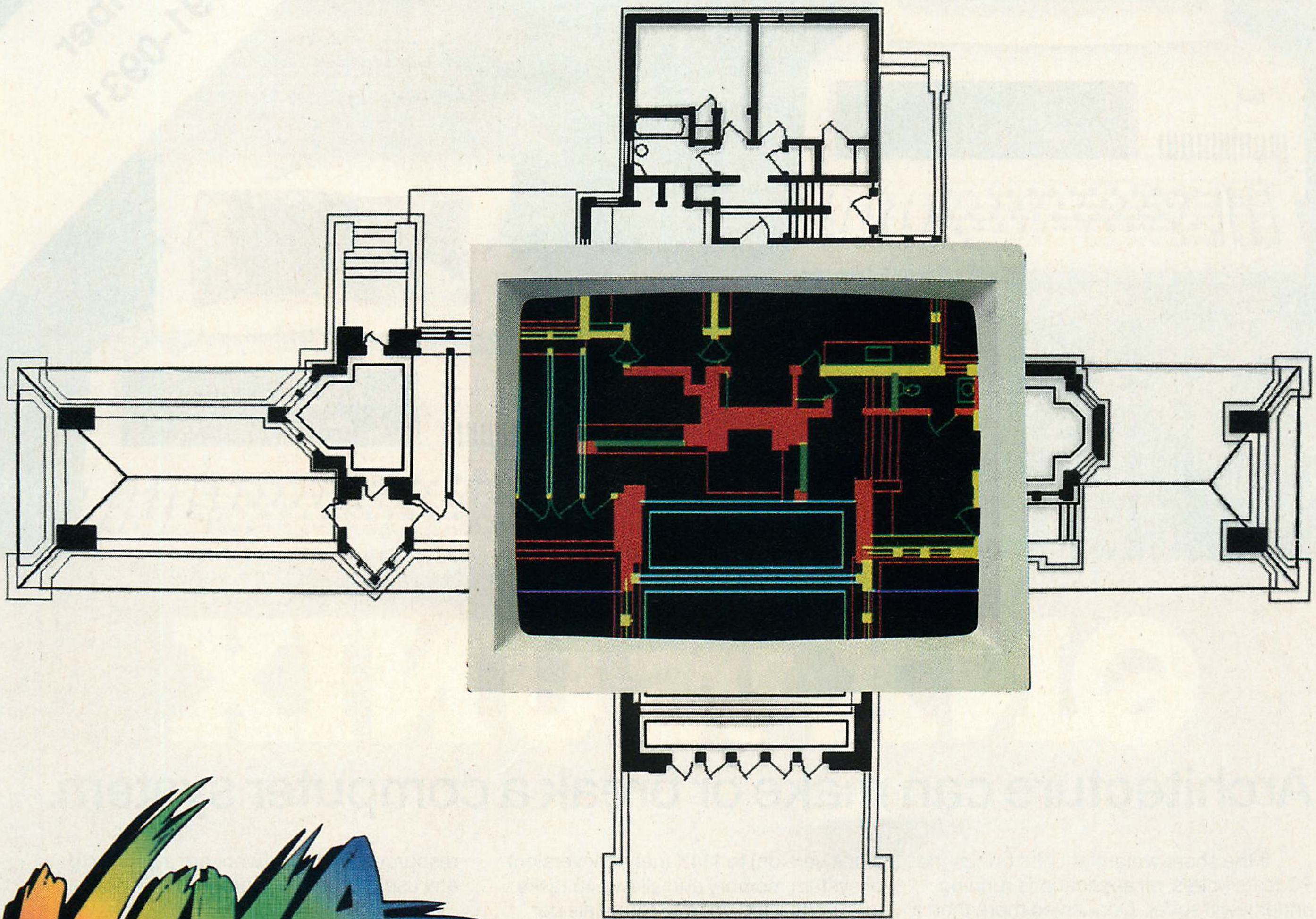
```
PI = 3.14159
Print PI
```

gives the answer of 3.141590118408203. Using double-precision variables yields the correct answer.

SUPPORT—OR LACK OF IT

In the course of testing for this review, I placed calls to both Borland and Microsoft to inquire about various

No Zoom Required!



VIVA

VIVA 2000 Series

It's all available, as you need it - INSTANTLY!

Over 5 times more information is instantly available in the 2048 x 2048 frame buffer than in any other standard high-res graphics board; up to 20 times more than your standard EGA. Just pan freely in real time over the entire drawing area.

Load your CAD drawing faster too. There's no waiting while extensive display lists are created since display lists aren't necessary.

No one else can save you as much time, and at the same time give you as much information.

It looks great, too.

Your drawing is displayed at the highest possible resolution 1280 x 1024 on any 19'' analog **non-interlaced** monitor with 16 bright vibrant colors to choose from.

Special CAD Features.

Enhanced AutoCAD[1] and VersaCAD[2] drivers are included with such features as smooth pan and scroll, instant hardware zoom, and pan freeze. A Microsoft Windows driver is also incorporated for your business needs.

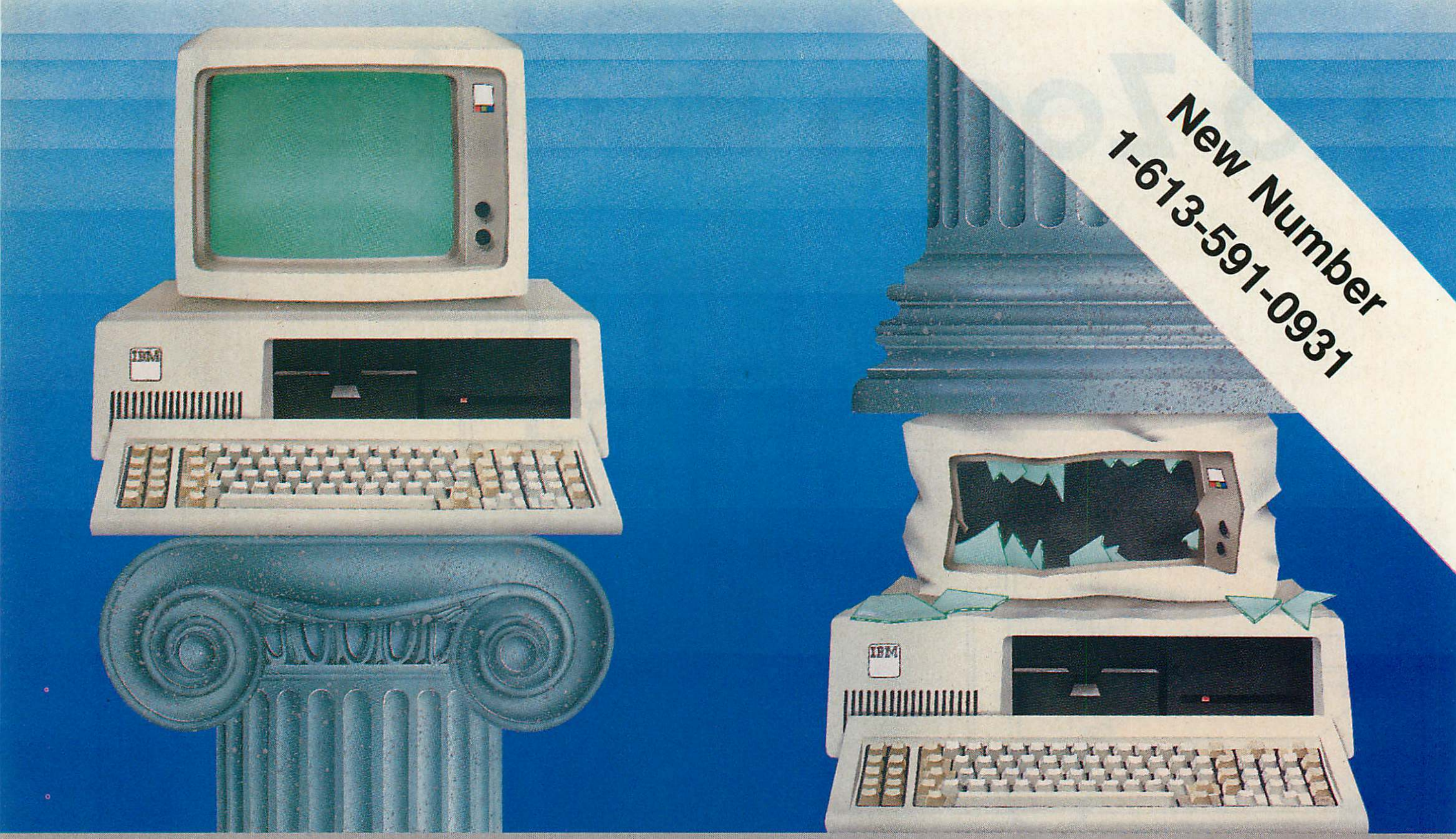
All of these advantages translate to increased productivity for you. So if you're interested in increasing your productivity, select the price/performance leader in graphics boards for CAD.

To find out more, Call (216) 464-6600.

QDP computer systems inc.
23632 Mercantile Road
Beachwood, Ohio 44122

CIRCLE NO. 193 ON READER SERVICE CARD

[1]AutoCAD is a registered trademark of Autodesk Inc.
[2]VersaCAD is a registered trademark of VersaCAD Corporation.



QNX vs UNIX

Architecture can make or break a computer system.

If the sheer weight of UNIX brings the PC to its knees, all applications running under it will suffer. Conceived more than a decade and a half ago, UNIX is today the result of modifications, additions and patches by hundreds of programmers. It needs the resources of at least an AT.

Compare this to the QNX O/S, designed by a dedicated team with a common purpose and complete understanding of both the software and the environment in which it must run. Having elegantly solved the problem of inter-task communications, QNX is more than capable of both networking and real time performance—the superior choice for process control and office automation systems.

Quick and efficient on a PC, QNX soars on an AT. QNX occupies 80K (stand-alone version) to 114K (network version) of system memory and allows 40 tasks (programs) and up to 16 terminals per computer.

QNX modular architecture facilitates easy adaptation and extensions by software developers for specific requirements. In addition, PC-DOS runs as a single-tasking guest operating system under QNX. With the DOS Development System, DOS EXE files can be developed in shorter time than under DOS itself.

Communication among all tasks is via "message-passing." Tasks anywhere on a network of up to 255 computers communicate rapidly and transparently with each other.

With the true distributed processing and resource sharing of QNX, all the resources on the network are available to any user. Application programs and data can be distributed over the network without having to go through a central file server.

Network growth is fast and simple. If your disk becomes a bottleneck, add a disk anywhere on the network. If your needs outgrow your present configuration, just add terminals and/or computers as required, without having to re-write programs and without system degradation.

If you would like to know the secret of the QNX architecture, please give us a call. We invite End Users, VAR's, OEM's and Software Developers to discover a whole new world of computing capabilities.

Over 30,000 systems have been installed worldwide since 1982.

THE ONLY MULTI-USER, MULTI-TASKING, NETWORKING, REAL-TIME OPERATING SYSTEM FOR THE IBM PC, AT, THE HP VECTRA, AND COMPATIBLES.

Multi-User	10 (16) serial terminals per PC (AT).	**C Compiler**	Standard Kernighan and Ritchie.
Multi-Tasking	40 (64) tasks per PC (AT).	**Flexibility**	Single PC, networked PC's, single PC with terminals, networked PC's with terminals. No central servers. Full sharing of disks, devices and CPU's.
Networking	2.5 Megabit token ring. 255 PC's and/or AT's per network. 10,000 tasks per network. Thousands of users per network.		
Real Time	2,800 task switches/sec (AT).	**PC-DOS**	PC-DOS runs as a QNX task.
Message Passing	Fast intertask communication between tasks on any machine.	**Cost**	From US $450. Runtime pricing available.

QNX

For further information or a free demonstration diskette, please telephone (613) 591-0931.

Quantum Software Systems Ltd. • Kanata South Business Park • 175 Terrence Matthews Crescent • Kanata, Ontario, Canada • K2M 1W8

CIRCLE NO. 181 ON READER SERVICE CARD

problems and to determine the level of support provided by each vendor. Neither vendor offers a toll-free number. Microsoft's support number is buried in the README.DOC file on the compiler disk, and Borland's had to be obtained from directory assistance.

When Borland's main switchboard finally answered after five tries, the operator suggested calling back in 30 minutes because all support personnel were busy. Borland would not return the call. Finally connected to the technical-support line, I waited six minutes before being assisted by knowledgeable and frank people who solved most problems on the spot, or at least suggested work-arounds.

Microsoft's technical-support line is answered by a recorded voice that instructs the caller to branch through several aural menus by using the telephone's touch-tone buttons. When I finally reached the correct extension for QuickBASIC support, another recording informed me that the company was not yet open for business and then disconnected. When I called back, I had to work through the touch-tone menus again and then wait six minutes for a human to answer. The level of support was somewhat less satisfying than Borland's, and all too often the response was "I'll find out and get back to you."

To test the level of Microsoft's support, its support personnel were asked the hypothetical question: why is it not possible to declare more than 15 dimensions of the form A%(1,1,1 . . .)? The answer should be that when the default OPTION BASE is 0, this declaration defines 2 elements on each dimension. Fifteen dimensions with 2 elements each equals 32,768 elements; with 2 bytes per integer, that is 64KB, the limit on array size. The support person did not know the answer, but promised to check into it and call back. A month has passed with no call.

Borland support was somewhat more satisfying than Microsoft's. Two problems were directed to Borland technical support regarding Turbo BASIC. In the initial copy of the compiler, a DRAW command string with two or more VARPTR$ functions did not operate properly. No compile-time or runtime error was generated, but the picture just did not get drawn correctly. Borland was aware of the problem and already had a corrected version available. This new version, 1.0c with files dated 4-20-87, is available to users upon return of the original compiler disk (a return authorization number must be obtained from Borland). Borland suggested a workable if not entirely satisfactory work-around.

The other problem involved appending data to text files. Turbo BASIC programs can append successfully to files created by other Turbo programs, but not to files created by Microsoft languages. When appending to the latter, Turbo programs do not remove the existing end-of-file marker (Ctrl-Z) before writing the new data. The resulting file could not be successfully read in its entirety because it always stopped when the Ctrl-Z was encountered at the end of the original data. Borland acknowledges the problem and promises a fix in the next release, but in the meantime, this could be a compatibility problem. Incidentally, QuickBASIC and BASICA have no problems appending to files created by Turbo BASIC.

DOCUMENTATION DIFFERENCES

Both compilers come with thick paperback manuals with glossy covers and bindings that rebel against staying open to the desired page without an anvil to hold it there. Purchasers of either system can count on an hour or so going through the README.DOC file and correcting the accompanying documentation. More errata were listed in the QuickBASIC document than in Turbo BASIC's; perhaps this is because Microsoft's product has been around longer.

The 465-page Turbo BASIC manual has much to recommend it, especially to nonprogrammers. It includes a short DOS primer as well as discourses on variable types, file types, and data storage techniques. Borland does not take the user's knowledge for granted, but some problems exist, including incorrect or fuzzy directions for the use of VARPTR$. The overall type size, especially in the examples, is too small.

A significant plus for Borland is that it provides sample programs on diskette that illustrate some interesting programming techniques, including a fine example of recursive programming applied to a sorting algorithm. The infamous Tower of Hanoi problem is neatly solved in a fascinating sample program that graphically illustrates the solution. A minispreadsheet called MicroCalc illustrates modular programming. It is not Lotus 1-2-3, but it is interesting to study.

The QuickBASIC documentation consists of a 595-page manual for version 2.0 to which a 70-page version 3.0 update has been prefixed, causing some discontinuities. Commands are contained in two alphabetic listings, each with its own index. Combine this

split organization with the 12 pages of README file and the result is far from ideal. For example, all references to files BCOM20.LIB, BRUN20.LIB, and BRUN20.EXE are to be changed to Bxxx30.xxx or Bxxx3087.xxx depending on whether the math coprocessor version of the compiler is installed.

The documentation has several inconsistent statements; the length of subprogram names is variously given as 31 and 40 characters; the reader is also told that programs compiled with the coprocessor version will run on machines without math coprocessors and then, on the next page, told that they will run only on machines with coprocessors. Which is correct? Neither and both, depending on options selected.

The QuickBASIC manual presumes that the user is an experienced programmer familiar with DOS and the process of compilation. This is especially apparent in the instructions for the linker, which would benefit greatly from examples. Throughout, it does little hand-holding. On the positive side, the graphic design is excellent.

FACE-OFF RESULTS

The face-off held within these pages did not produce a clear winner. Both of these BASIC offerings are excellent values and represent a quantum improvement over BASICA in both speed and ease of programming.

Turbo BASIC continues the Turbo Pascal tradition with ease of installation, lightning-quick compilation speed, and overall nimbleness of operation. It is especially recommended for programmers graduating from interpreters.

Turbo BASIC's context-sensitive help, well-executed manual, sample programs, and technical support are a significant cut above the competition. For serious development work, Borland's product is unsurpassed for working fast and small—in other words, for the very kind of development that the BASIC language is best suited. Its recursive abilities can produce compact and elegant code.

However, like its fellow Borland language, Turbo Pascal, ultimately it is unsuited for large-scale work, where it is hampered by a primitive debugger, the need for manual segmentation control, and especially the lack of support for separate compilation. Over the long term, using INCLUDE statements as the only means of modularization becomes severely limiting.

Microsoft's QuickBASIC is a full-blown, powerful compiler package that is more suited for the experienced programmer developing large commercial applications. (Neither Borland nor Microsoft charges royalty fees for commercially distributing products produced with these compilers.) Although its data space is smaller than Turbo BASIC's, it offers an excellent debugger, modularization at the object-code level, and overcomes many of the traditional limitations of BASIC as a serious language. QuickBASIC's documentation and overall complexity might be somewhat intimidating, however. This complexity makes it unwieldy in operation and defies the underlying concept of the BASIC language. Such is the price of power. Users of previous versions of QuickBASIC should convert to version 3.0 without delay.

Although Borland and Microsoft are competing head-to-head in the area of BASIC compilers, their respective products are ideally suited to different segments of that market. It is the intended scope of applications that must determine the selection.

Justin Crom is a chemical engineer working as manager of economic evaluation for a major oil company. He has almost 20 years of experience with computers, much of it in developing engineering and decision support applications in high-level languages.

The only quality multi-scan monitor that's priced for business and the mass market that can be sold profitably.

What does this mean to you?

Easier sales. Faster sales. More profitable sales.

And best of all, the Teknika MJ-503 Multi-Scan Color Display Monitor is compatible with IBM PC/XT/AT, their clones and virtually any other computer made today.

Its wide frequency range of 15kHz to 34kHz automatically scans for CGA/EGA/PGA. Other features include 926 x 580 pixels, 16/64/ALL color switch, TTL/ANALOG input signals, text SW (green), and the interface cable is included.

The barriers are down...and your profits will be up. Don't delay, contact your Teknika representative today.

Exclusive 2-Year Limited Warranty

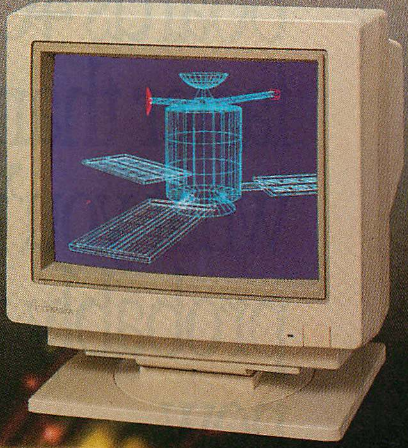
The longest and strongest warranty in the industry. Covers parts, CRT and labor.

Non-Glare 13″ Screen

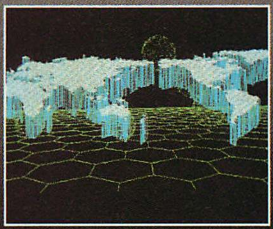
Less eye strain, easier to read.

Easily Accessible Controls

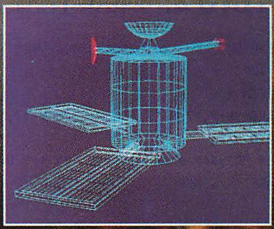
Controls are up-front to allow for easy adjustment of horizontal position, brightness and contrast.



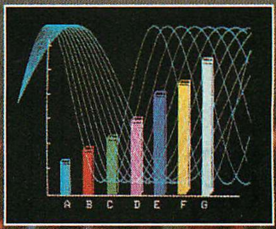
OPTIONAL SWIVEL BASE



EGA



PGA



CGA

TEKNIKA BREAKS THE PRICE BARRIER ON MULTI-SCAN MONITORS

EXCLUSIVE **2-YR.** LIMITED WARRANTY

TEKNIKA

TEKNIKA ELECTRONICS CORPORATION 353 Route 46 West, Fairfield, New Jersey 07006 ■ 201-575-0380
A Subsidiary of *FUJITSU GENERAL LIMITED* 1-800-835-6452 Display Monitor Division

CIRCLE NO. 138 ON READER SERVICE CARD

LISTING 1: SCRNTEST.BAS

```
'FILL SCREEN WITH TEXT
defint a-z
a$ = string$(72,"0")
start! = timer

for i=1 to 10
  cls
  for r = 1 to 24
    print a$
  next r
next i
secs! = timer - start!
print secs!
end
```

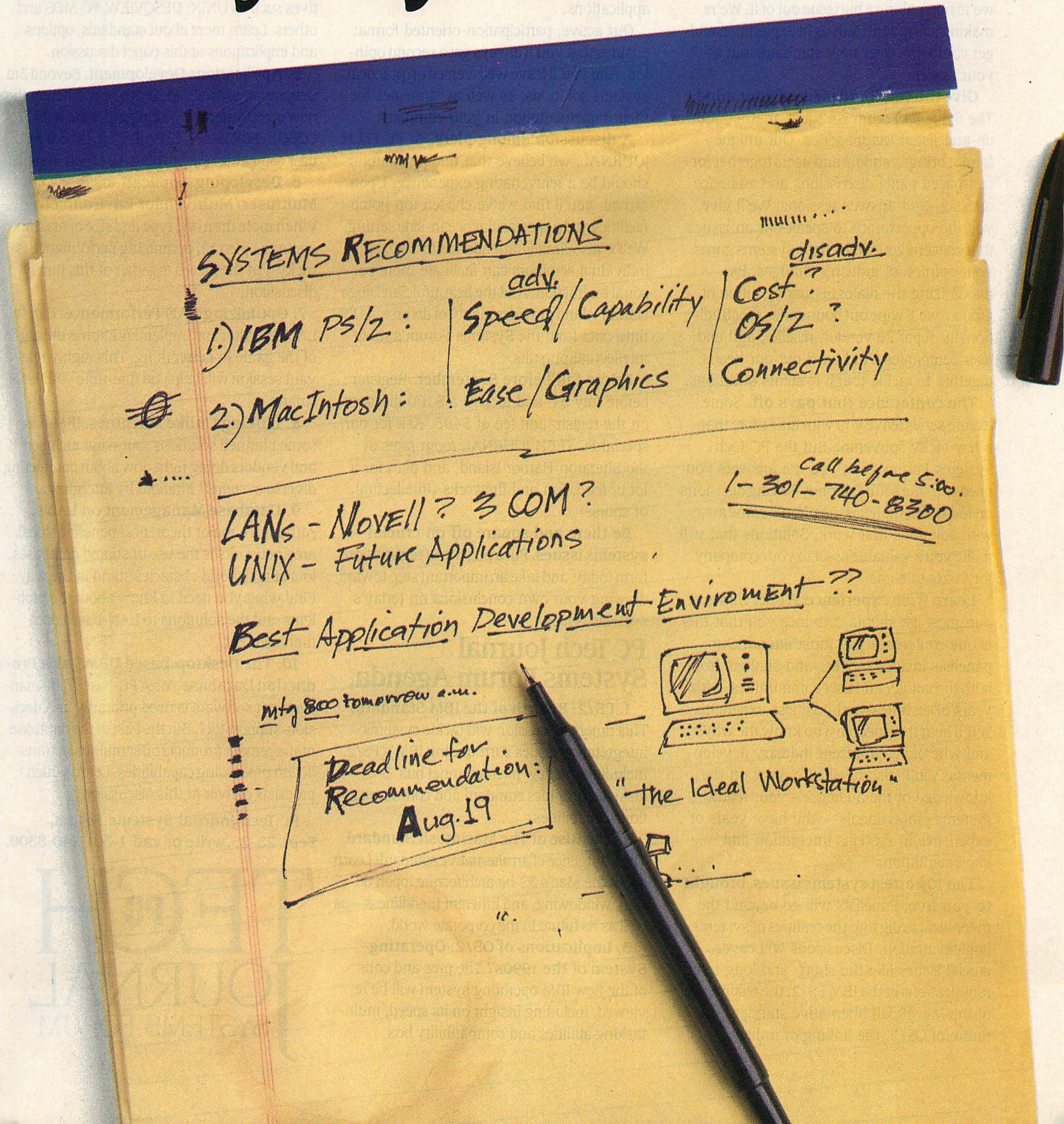
LISTING 2: DRAWTEST.BAS

```
5 ' ADAPTED FROM IBM BASIC MANUAL VER 3.0, PG. 78
10 KEY OFF
20 SCREEN 1,0
30 STAR$="M+7,17 M-17,-12 M+20,0 M-17,12 M+7,-17"
40 SCALE = 0
50 S$ = "C1;S=" + VARPTR$(SCALE) + " BM-2,0 X" + VARPTR$(STAR$)
60 START! = TIMER
70 '
80 ' Time empty loops
90 T1! = TIMER
100 FOR I = 1 TO 50
110   FOR SCALE = 1 TO 40 STEP 2
120   NEXT SCALE
130 NEXT I
140 T2! = TIMER
150 '
160 ' Start the test
170 FOR I = 1 TO 50
180   CLS : DRAW "BM300,25"                          'INITIAL POINT
190   FOR SCALE =1 TO 40 STEP 2
200     DRAW S$
210   NEXT SCALE
220 NEXT I
230 SECS! = (TIMER - T2!) - (T2! - T1!)
240 PRINT SECS!
250 INPUT "PRESS ENTER TO QUIT"; X
```

LISTING 3: SAVAGE.BAS

```
10 Rem     SAVAGE BENCHMARK
15 Rem        Adapted from BYTE magazine, Vol. 10 No. 11
20 rem
30 defint i
40 defsng a
50 print "Calculating................";chr$(7)
60 a=1!
65 start = timer
70 for i=1 to 24999
80   a = tan(atn(exp(log(sqr(a*a)))))+1!
90 next i
95 seconds = timer - start
100 print chr$(7); a, "Time ="; seconds
110 end
```

If you're having trouble drawing conclusions on today's systems issues,



Draw on the expe

The PC Tech Journal Systems Forum In San Diego: a meeting of the minds you won't want to miss. Recent changes in the computer industry raise pressing questions for all of us. But at PC TECH JOURNAL, we're not making a big issue out of it. We're making it an event. Join us in September, and get confusion over new standards out of your system.

Give the vendors a piece of your mind. The Systems Forum isn't just another shut-up-and-listen lecture series. Our unique format brings vendors and users together for fast-paced panel discussions and audience question-and-answer sessions. We'll give you the opportunity to speak out on issues that concern you, and tell top systems gurus your opinion of industry changes. By maximizing the dialogue among different groups, we'll wipe out boundaries that traditionally separate vendor, reseller, and end user companies. And we'll put our heads together to tackle tough systems questions.

The conference that pays off. Some lecture series leave you with little more than a few tacky souvenirs. But the PC Tech Systems Forum gives you the answers you need to make sound, money-saving decisions on today's complex issues. You'll come away with solutions that work. Solutions that will make you a valuable asset to your company for years to come.

Learn from experience. A brief glance at our guest list should convince you that this is one conference you must attend. Our panelists include movers-and-shakers from both the vendor and user communities, with years of technical and business experience. You'll hear from people who know the whys and wherefores of recent industry developments. You'll also be able to draw on the knowledge of the audience—your fellow systems professionals—who have years of experience in systems integration and implementation.

The 10 hottest systems issues, brought to you live. Panelists will go beyond the theoretical to dig into the realities of systems implementation. Discussions will cover crucial issues like the short- and long-term ramifications of the IBM PS/2, the resurgence of the Macintosh alternative standard, the future of OS/2, the linking of unlike machines, and optimizing LAN performance, to name a few. Other issues to be examined are building expert systems, alternatives to DOS, applications development, and turning data management software into production applications.

Our active, participation-oriented format ensures that you'll always get a second opinion. And you'll leave well-versed on practical systems solutions, as well as strategies for their implementation in your company.

A discussion among piers. At PC TECH JOURNAL, we believe that finding solutions should be a rejuvenating experience. Upon arrival, you'll find we've chosen top-notch facilities and a spectacular ocean-side setting. We've also limited attendance to 500, so individual seminars can facilitate more personal interaction. And the beautiful San Diego Harbor Island provides plenty of diversions for time-outs from the Systems Forum's more intellectual pursuits.

Prices fall before September. Register before July 31, and receive a $100 discount on the registration fee of $495. Ask for our special PC TECH JOURNAL room rates at the Sheraton Harbor Island, and plan for a lot of food, fun and fireworks (intellectual, of course).

Be there and square off on crucial systems issues. Put a pencil to a registration form today, and take an important step toward drawing your own conclusions on today's systems issues.

PC Tech Journal Systems Forum Agenda.

1. PS/2: Rebirth of the IBM Standard. This timely discussion will tackle systems integetation issues surrounding IBM's PS/2, including the new microchannel bus, 3½" diskette, graphics standard and communications capabilities.

2. The Rise of The Macintosh Standard. The resurgence of an alternative standard: Learn about the Mac's 32-bit architecture, open design, windowing, and inherent friendliness—as well as its future in the corporate world.

3. Implications of OS/2: Operating System of the 1990s? The pros and cons of the new IBM operating system will be reviewed, including insight on its speed, multitasking abilities and compatibility box.

4. Waiting for OS/2: Alternative Operating Systems. A new generation of multitasking/ multiuser operating systems is arriving for the PC. OS/2 is announced but undelivered. It's appropriate to examine alternatives such as UNIX, DESQVIEW, PC-MOS and others. Learn more about standards, options and implications at this panel discussion.

5. Applications Development. Beyond 3rd Generation towards AI: Learn how different the new object-oriented/AI languages are from COBOL, FORTRAN, C, BASIC, etc., and how user companies are building AI/expert systems.

6. Developing Applications in a Multiuser/Multivendor Environment: When more than one type and size of machine are in the picture, optimizing performance is tricky. Put the pieces together at this timely discussion.

7. Optimizing LAN Performance. Careful LAN selection and implementation is the name of the game in networking. This highly relevant session will help you fine-tune your LAN smarts.

8. Linking Unlike Machines. IBM has some planned solutions, but what are users and vendors doing right now about connecting diverse systems? Find out by attending.

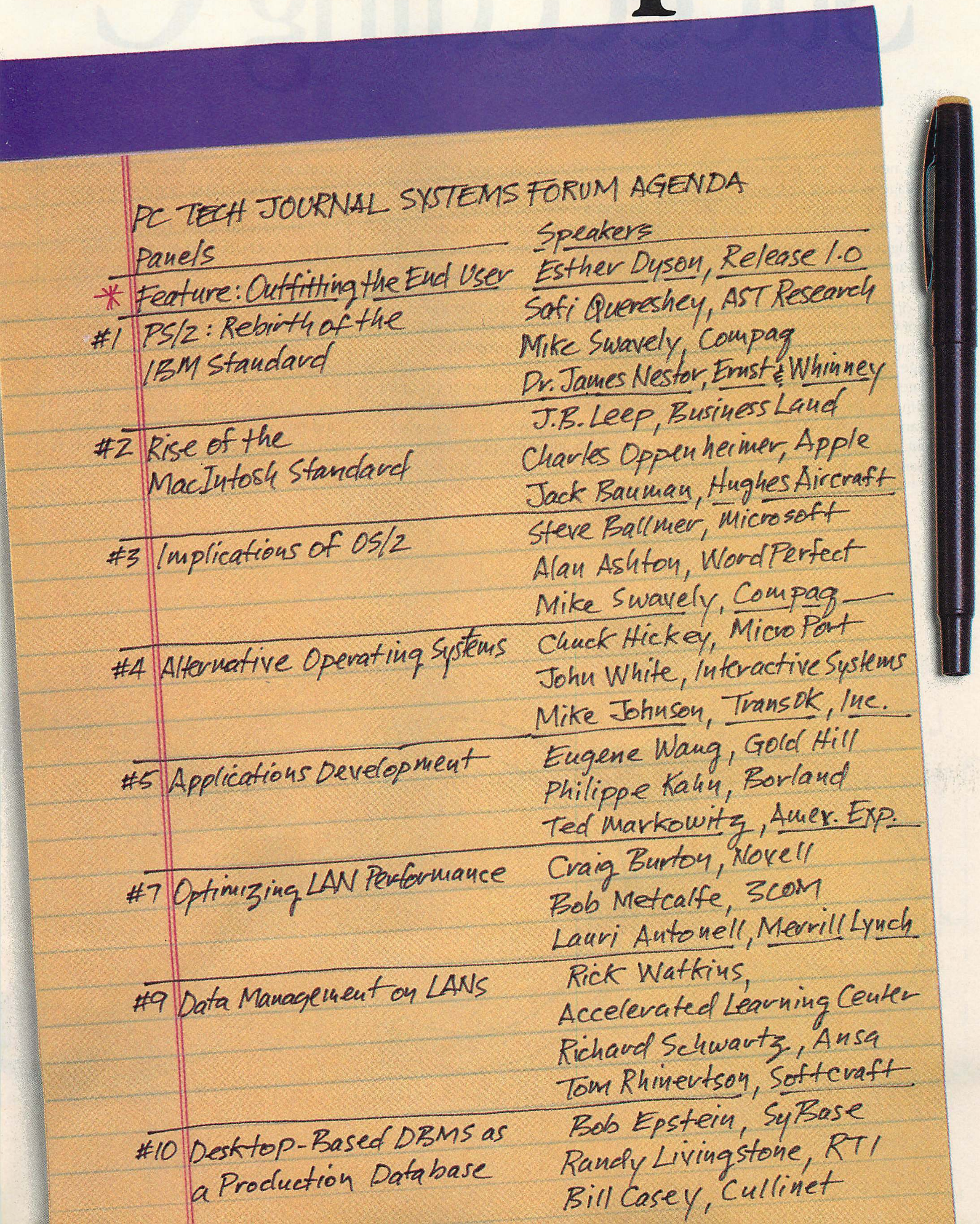
9. Database Management on LANs. Although a major theoretical benefit of local area networks is the use of shared databases, lots of real world obstacles stand in the way. Find what you need to know about the problems and the solutions to LAN-based data management.

10. The Desktop-based DBMS as a Production Database. Most PC-based data management software is used primarily as a decision-support tool. But the best of the database management products offer multiuser, transaction processing capabilities. Learn which products deliver in this discussion.

PC Tech Journal Systems Forum, Sept. 23-25, write or call 1-301-740-8300.

PC TECH JOURNAL
SYSTEMS FORUM

rience of experts.



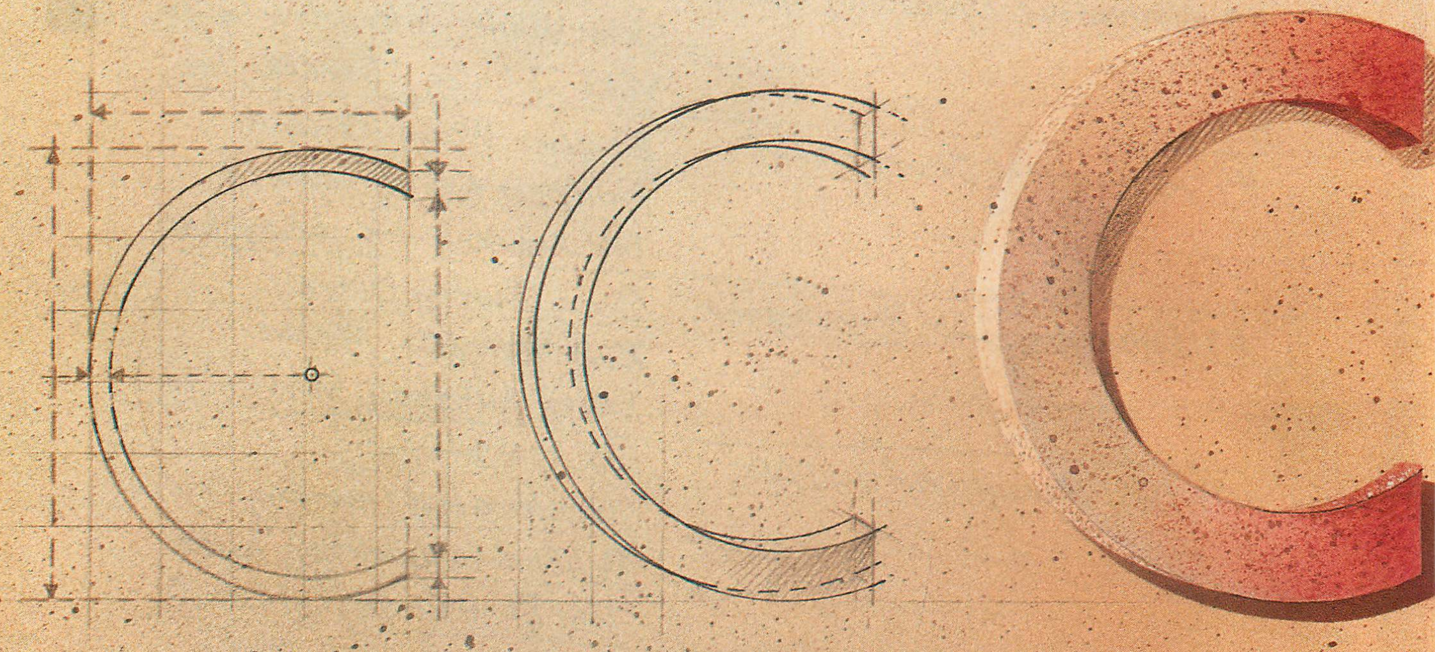
Succeeding C

The new C++ programming language is touted as being a better C. It was designed to make life easier for the programmer by adding a rigid programming discipline as well as other features to the original C language. How well it delivers on this design is open to question. The potential utility of C++ is considered here, as are two compilers that implement the new language: Advantage C++ from Lifeboat Associates and Guidelines C++ from Guidelines Software.

Standard C was designed to be an efficient, portable, systems programming language. It used the computer's underlying hardware and relaxed type checking to make programming convenient for the experienced professional. At the time the standard C programming language was created, this approach was acceptable, given the relatively modest computers upon which C and UNIX were first developed and the small number of programmers typically working on a program.

As the demands of increasingly complex projects and larger programming teams revealed the shortcomings of standard C, a new language was needed to allow the programmer to create and manipulate software objects more easily. Today's larger projects require a rigid programming discipline and added language features.

Along with structured programming, *object-oriented programming* has evolved to make large software systems easier to maintain and, thus, more cost effective. The basic concepts of object-oriented programming—message passing, data abstraction, and class inheritance—permit the programmer to write programs as groups of small, easily reused "objects" that encapsulate data and procedures. These new concepts of program design require new constructs in programming languages.



C++ adds enhancements and object-oriented programming to standard C, but is it really better? C and C++ are compared, and two C++ compilers are considered.

MARTY FRANZ

As defined in Kernighan and Ritchie's classic, *The C Programming Language* (Prentice-Hall, 1978), C does not include object-oriented constructs. Although a gifted programmer can implement the idea of object-oriented programming in C, such constructs are difficult to achieve because of C's essentially loose, low-level nature.

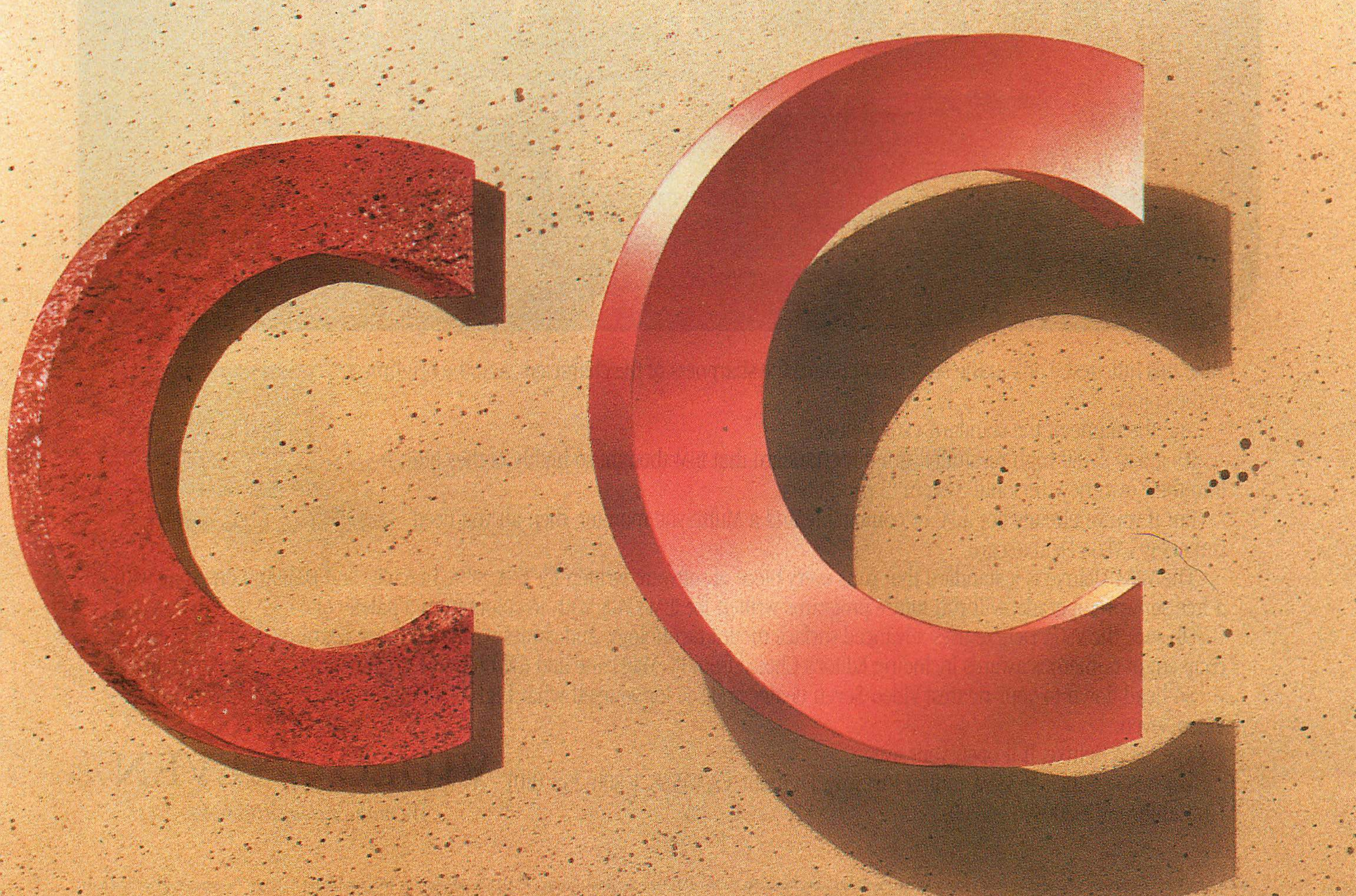
C++ not only offers object-oriented programming facilities, but also provides structured programming and features to correct other C shortcomings. Developed for the UNIX operating system by Bjarne Stroustrup at Bell Labs, C++ already has been proved successful there, with nearly 650,000 lines of code developed. *The C++ Programming Language* by Stroustrup (Addison-Wesley, 1986) is the comprehensive reference for the new language.

In recognition of the large amount of software that has been written in C, C++ was designed as a proper superset of C to ease software conversion. In fact, the C++ compiler is actually a preprocessor that emits C source code, which then is compiled using a standard C compiler. Normal C programs pass through the C++ compiler unchanged, allowing the vast base of C programs to be used in a C++ environment without conversion. To use the additional features of C++, however, a programmer must absorb a large amount of new information.

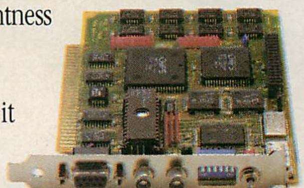
C++ ENHANCEMENTS

One enhancement offered by C++ involves function arguments. Standard C does not check for type matching between the function's declaration and invocation, or for the number of arguments in a function. This loose arrangement, although occasionally convenient for an expert, can cause unintended data truncation or lengthening and is a common source of bugs.

ILLUSTRATION • NENAD JAKESEVIC/SONJA LAMUT



How to see VGA for what it really is.



Take this little test. Look at the screen and notice the sharpness of the characters. And the brightness of the colors.

And the name of the standard. VEGA Deluxe.™

It's a 640 x 480 high resolution graphics standard that IBM thought so highly of they built it into their own new Personal System/2 computers.

But if you've already got a PC or compatible and a MultiSync monitor, then all you need is the $495 VEGA Deluxe card.

The VEGA Deluxe is a standard that runs every piece of software written for CGA, MDA, Hercules and EGA. And one that, with a free software upgrade, will run future programs written for IBM's VGA. With 16 colors from a palette of 64.

Oh sure, the VEGA Deluxe uses advanced surface mount technology. And comes with a two year warranty. And has been the recipient of numerous awards including Editor's Choice from PC Magazine and the Ultimate Machine from PC Tech Journal.

So head down to your nearest Video Seven dealer and see for yourself what high resolution really looks like. Or contact Video Seven.

The VEGA Deluxe. It'll open your eyes.

Video Seven Inc., 46335 Landing Parkway, Fremont, CA 94538, 1-800-238-0101 (in California, 1-800-962-5700).

VIDEO SEVEN

We make a clear difference.

CIRCLE NO. 204 ON READER SERVICE CARD

C++ rigidly checks arguments for type and number, forcing conversions to the calling argument's type if necessary. It also extends C's **extern** key word so that an external function declaration contains its calling list, much as the proposed ANSI C standard has done. (Although the syntax is different, C++ vendors plan to adopt the ANSI syntax when it is finalized.) Placing **extern** in an **#include** file ensures correct type checking across multiple source files while still allowing independent compilation to occur.

Along with this rigorous argument checking, C++ also can pass a variable number of arguments to a function and assign default values. This long-overdue improvement, which also is present in the ANSI standard for C, allows generic formatting functions, such as **printf()**, to be written cleanly.

The C++ programming language uses ellipses to declare a function with an unspecified number of arguments:

```
int printf(char* ... );
```

It specifies default arguments by adding an = followed by a constant after the argument's type declaration in the function's header:

```
int myfunc(int, int =0);
```

In this example, both arguments to **myfunc** are integers; the second argument defaults to zero if it is not supplied by the programmer.

C++ also extends C's normal declaration syntax to include references, which point to a data type's address rather than to its value. In C, passing to functions by reference is required with large data structures that cannot be passed by value. References are simulated in C by using pointers to data, **typedefs**, and **macros**, an approach prone to errors. In the C++ language, references allow these types of data to be passed without cumbersome pointer notation. For example, in

```
bigstruct a; //large structure
bigstruct& b = a;
```

the identifiers **b** and **a** represent the same data, with **b** defined as a pointer to **a**. The notation // is one way to indicate a comment line in C++. All characters that follow // are ignored up to the end of the line.

Another enhancement, the **inline** function type, is of particular interest to those PC programmers who must contend with limited stack requirements. It allows functions to be compiled directly into the calling module at the point where they are called, which results in faster execution and reduced call overhead. However, unlike C preprocessor macros, **inline** has normal argument binding, which helps to avoid those often-subtle side effects.

OBJECT-ORIENTED

Although the enhancements just discussed are convenient, they are not as important as the object-oriented programming statements in C++. All of these facilities are based on the **class** statement, which describes a new object type. Although similar to C's **struct** statement in syntax, **class** does more than specify data layout in terms of simpler types; it also includes initialization and clean-up, class-related data manipulation, and coercion functions that are called automatically when converting a data type to a class type.

C++ can pass a variable number of arguments to a function and assign default values—an improvement that is long overdue.

To determine how a class is used in C++—and why it is an important extension to the C language—data structures and defined types in conventional C should be examined. A standard C data structure is defined in *The C++ Programming Language* as

```
struct intset {
  int cursize, maxsize;
    int *x;
};
```

This function defines a structure called **intset** (a set of integers) that contains the elements **cursize**, **maxsize**, and **x**. The variable **x** holds a pointer to dynamically allocate storage for the set. Although C's **intset** is said to undergo a rigorous check when used in a program, problems still arise when using this method to encapsulate data.

In the first place, the contents of the data structure are not private—that is, any function using **intset** can access the member data items within **intset**. These functions depend entirely upon the **intset**'s internal structure. If this data structure were to change, and, for example, were to use **long** instead of **int**, every function that accessed **intset** also would have to change. The standard C language compiler would not catch this particular problem—it would be up to the individual programmer to track down and correct each one.

Second, simply declaring a variable to be an **intset** does not guarantee it will operate correctly as one. Initialization may be needed, or, at the very least, allocation of automatic storage for the variable's contents using the function **malloc()**. The **struct** declaration does not insure that, once defined, a variable will be used correctly.

These two problems, although trivial in small programs, can be nightmarish for the programmer maintaining a large software system that has many defined types in dozens of modules. Improved protection of a data item's structure and more discipline regarding its usage are required.

The C++ **class** statement comes closer to solving these problems than the **struct** declaration does. A class for **intset** is defined as

```
class intset {
  int cursize, maxsize;
  int *x;
public:
  intset(int m, int n);
  ~intset();

  int member(int t);
  void insert(int t);

  void iterate(int& i)
    {i = 0;}
  int ok(int& i)
    { return i < cursize; }
  int next(int& i)
    { return x[i++]; }
};
```

This definition syntactically resembles a **struct** in that braces are used to enclose the contents. However, the actual operation of a class is somewhat different. First, a class statement contains both functions and data. The function declarations, called *member functions*, are the only functions allowed to directly access data in a class statement.

Second, the label **public:** separates the parts of a class statement that are visible to the outside program from the parts that are strictly private. The public functions and data items are the programmer's sole external interface to a class statement. The C++ compiler protects data items preceding them from direct manipulation by functions outside the class. This isolation prevents functions outside the statement from changing if the data and functions inside change—a vital way to simplify the maintenance of the final program.

The first function, **intset()**, which bears the class's name, is known as the *constructor*. Whenever a new member is instantiated (that is, brought into existence, with the identifier beginning its scope) by a declaration, the constructor is called in order to allocate storage or perform initialization. The second function, **~ intset()**, known as the *destructor*, is called automatically whenever an object of class **intset** is no longer in scope; it releases any dynamic storage that was allocated by the constructor. The statements **new** and **delete** in these functions allocate memory for their objects, just as **malloc()** and **free()** do in normal C. Unlike **malloc()** and **free()**, however, **new** and **delete** automatically allocate one object of correct size without casting the result or multiplying by **sizeof()**. C++ helps ensure correct use of class statements by requiring that constructors and destructors be defined and called.

The remaining class functions perform various manipulations on the set of integers. Because the integers cannot be directly accessed outside the class-member functions, all the services that users require must be implemented through class-member functions. Therefore, **insert()** adds an integer to the set; **member()** determines if a given integer is a member of the set; **iterate()** starts a loop through the set; **ok()** checks for the next member; and **next()** accesses it. With these functions, the user can loop through the set of integers starting at the first element. Within the scope of a class, functions can be both defined and declared. The member functions **iterate()**, **ok()**, and **next()** all have been defined as part of the object; they require no additional function definitions.

The C++ Programming Language definition of **intset**, which creates and prints a set of random integers, illustrates the uses of **class** statements and objects in a complete program (see listing 1, INTSET.CXX). Although the program's source code still looks like C, some important differences exist.

First, the class statement defines the class of objects, or set of integers. Its data items consist of the set's size (**maxsize**), the number of elements currently in the set (**cursize**), and a pointer to the integers in the set (**x**).

Following the data declaration, the key word **public**: precedes a list of function declarations. Following the entire class declaration, some member functions of the class are defined separately. The function declarations are preceded with the name of the class and ::, as in **intset::member()**, because member functions are defined locally to the class. In a complete program, several functions could be named **member()**, one defined for each of the other class statements used in the program. This notation, while lengthy, allows functions to be defined in files separate from the class declaration, then compiled separately. Functions also can be defined locally, within the scope of the class declaration, as were **iterate()**, **ok()**, and **next()**.

To use a class, first a specific object belonging to it must be declared:

```
intset s(m, n);
```

This statement not only declares **s** to be a member of class **intset**, but also calls the constructor to allocate memory for the object's data. The arguments **m** and **n** are passed to the constructor for dynamic memory allocation and other initialization procedures.

Like other function calls in the C++ language, this constructor call is rigorously checked during compilation for type and number of arguments. Because forgetting to initialize dynamic data structures is a common programming error to make, the constructor is another way that C++ increases the reliability of large programs.

After an object has been instantiated, the functions that make up the class can be called by using the object's name as a selector, as in

```
s.insert(t);
```

or as a pointer, as in

```
set→iterate(var);
```

Because this can be hard to read, C++ extends normal C operators (such as +, −, and even [] for subscripting) to work with objects of various class statements. This *operator overloading* facility enables the seamless use of class statements in programs, exactly as if they were basic types, such as **int** and **float**. The statement

```
cerr << "set: " << s << "\n";
```

uses the dyadic operator **<<** (normally left shift) to write variables and string literals to the stream file **cerr**; s is a character array in this example. This statement is equivalent to

```
printf(cerr, "set: %s\n", s);
```

in the normal C language.

A function that implements an overloaded operator is declared by naming it **operator** followed by the particular operator. Any of the C operators, except for →, . , . , , and ?: , can be used this way. For example,

```
istream& operator >>(char*);
```

declares the function **operator >>**, which accepts a pointer to characters. When the operator **>>** is encountered in a statement, this function is called if the argument types match the declaration. Additional functions also can be declared with the same name, but each takes a different argument type:

```
istream& operator >>(int&);
istream& operator >>(char&);
```

These functions print integers passed by reference and single characters. The C++ compiler determines which function to call based on the type of argument used in the invocation. Like Ada, any C++ function can be overloaded by uniquely distinguishing the argument types passed to it.

Operator overloading should be considered carefully before using them for user-defined types. For example, overloaded operators are not inherently commutative. A scalar-addition vector operator that is defined as

```
vector operator + (vector v, int i)
```

cannot be invoked by the expression **v1 = 2 + v2**. Also, **v1 = v1 + 1** is not the same as **v1++** unless **operator++** is appropriately defined. Overloading an operator in an obscure manner can do more harm than good.

Isolating data in a class as to their member functions often is a problem. For efficiency, some functions could require direct access to the private data of two or more class statements. To bypass this restriction, C++ allows class statements to specify *friend* functions that are not strictly members of a single class, but still have access to the statement's private data. The particular function is declared to be a friend in the class declaration.

When derived classes are used, the programmer can define functions in the base class that will work on objects of all derived classes.

Another restriction is that class statements sharing a similar structure must still be declared independently. C++ allows one statement to share the facilities of another using a mechanism called *derivation*. The declaration of a class called **room** might be

```
class room {
  int length, width, height;
  char *roomname;
public:
    // . . .
}
```

Another class, **living_room**, can be derived from **room** using the syntax

```
class living_room :
  public room {
  int number_of_chairs;
public:
  // . . .
}
```

This declares that **living_room** uses the properties of **room**. Unless redefined, the data (length, width, height), constructor, destructor, and member functions of **living_room** are the same as those of **room**. Any replacements declared within **living_room** override those of **room** and are unique to **living_room**. This mechanism, called *inheritance*, allows extensive reuse of the program code by carefully choosing class definitions. In this example, **room** is said to be the *base class*, and **living_room**, a *derived class*.

A sample use of a derived class might be in a graphics module in which a base class contains the location, the color, and the number of sides for a given polygon:

```
class polygon {
  int sides;
  point topcorner;
  color fc, bc;
  // . . .
public:
  point where()
  { return topcorner; }
  void setcolor(color f, b)
  { fc = f; bc = b }
  // . . .
};
```

With this base class defined, other derived class statements can be defined that handle special cases:

```
class square:public polygon {
  int length;
  // . . .
public:
  void draw();
  void move(point to)
  { topcorner = to; draw(); }
  // . . .
}
```

This class is defined for squares, which have four regular sides of the same length. The functions **square::draw()** and **square()::move()** can take advantage of the square's symmetry when drawing or moving the object on a graphics screen, but the programmer still can use the functions **setcolor()** and **where()** of the **polygon** class, because these functions are not unique to squares. Therefore, the programmer must write and test only two new functions for this new class.

When using derived classes, the programmer can define functions in the base class that work on objects of all derived classes. This requires matching the function and object at runtime rather than at compile time, because the compiler cannot determine exactly which derived class is being used in a function call. This situation is analogous to using unions in standard C. To tell the compiler which data type is in use, the programmer includes a common "type" field in the union and determines what other fields in the union to use based on this value.

To avoid this practice, which is prone to bugs, C++ uses *virtual functions*, declared in the base class by the key word **virtual**. Derived classes can either redefine these functions when needed or use the version in the base class. A single pointer of overhead is

added when virtual functions are used within a class. At runtime, the pointer is resolved to the correct function for the class of the object for which it is invoked. This is the only instance in the C++ language where binding of functions and objects occurs at runtime, which reduces the runtime overhead of compiled C++ programs.

Thus, C++ can abstract data (because data in a class are private), pass messages (by invoking class functions with object selectors for specific objects), provide inheritance (using derived classes and virtual functions), and support limited runtime binding. These facilities are the building blocks of object-oriented programming.

Unlike some other object-oriented languages, such as Smalltalk, C++ performs very little extra processing at runtime. The compiler and linker perform almost all correspondence between objects and functions, and the object code is almost as efficient as standard C. As a result, some important object-oriented language facilities are missing from C++, including automatic garbage collection of allocated storage, multiple inheritance (where one derived class has two base classes), and concurrent programming. These have been omitted because they would have compromised compatibility with existing C compilation and linking, or would have added more overhead to standard C programs passed unchanged through the C++ compiler.

LOOK-ALIKE COMPILERS

Two C++ compilers are now available for DOS machines: Advantage C++ and Guideline C++. Both generate C code that can be compiled by the Microsoft C compiler. Advantage C++, from Lifeboat Associates (which also carries the Microsoft C compiler and other programming tools), is packaged in a PC-manual-sized binder with slipcase and two diskettes. In addition, a version is available that operates with the Lattice C compiler. Guidelines C++, from Guidelines Software, is packaged in a standard 8.5-by-11-inch loose-leaf binder, with two diskettes.

These compilers bear more than a casual resemblance. Because both are derived from the same AT&T source code used in UNIX implementations, portability between the PC and UNIX versions should be uncomplicated. This implementation method has some disadvantages. C++ is actually a preprocessor that translates its input C++ code and **include** files into C code; thus, lexical analysis, parsing, and

symbol-table maintenance must be performed twice—once by C++ and once by C. The resultant C file is typically large and generates a great deal of disk I/O. The performance degradation may not be noticeable on a minicomputer or mainframe UNIX system, but it is painfully evident on a PC.

Before C++ can be used, Microsoft C must already be installed on the system, requiring a separate purchase before either preprocessor can be used. Both Lifeboat and Guidelines include Stroustrup's book as part of the documentation. Both preprocessors make heavy use of DOS environment variables, which can create problems for users with early (2.x) releases of DOS, where environment size is limited to 160 bytes. The Microsoft C compiler utility SETENV can be used to patch COMMAND.COM to allow additional environment space.

Both also generate output in C code that can be debugged with Microsoft's powerful CodeView debugger. Because both use the Microsoft compiler and linker, they can interface with commercially available libraries, such as those by Greenleaf Software, Inc. or Blaise Computing, Inc. Except for virtual functions, data structures created in C++ classes have no hidden overhead such as links or pointers; they can be passed freely among C++, C, and assembly functions according to standard calling conventions.

After preprocessing, the C source generated for the program is rather dense, but otherwise compliant with the standard described by Kernighan and Ritchie. The lengthy header declarations contained in the **include** file, **stream.hxx**, have been omitted for clarity (see listing 2, INTSET.C).

As the listing shows, the translated C code makes heavy use of long identifiers to translate C++ references such as **intset::insert()** into legal C-function names, the **#line** preprocessor directive, and casts. In actual use, this C code is not maintained directly; instead, the C++ source is edited and preprocessed each time it is modified.

Currently, no cross-reference or LINT-like utilities on the PC directly support C++ source code. However, because C++ already checks function arguments, some of the key functions of **lint** have been incorporated into the language itself. Only Advantage accepts the Microsoft extensions to C—key words **near**, **far**, and **pascal**—that are used in Microsoft Windows. The Microsoft extensions are not portable to other environments such as UNIX.

From micro to mainframe, Casio's got the hardware for your software.

At first glance, it may look like an ordinary calculator, but the Casio solar CM-100 is anything but. It's an extraordinary software tool that's as useful in programming an Apple™ as it is a mainframe IBM.™

The key to the CM-100's incredible flexibility is Casio's adjustable bit-size selector which can be set to suit any size computer up to 32 bits. And its block display which can, by scrolling blocks of 8 digits at a time, display up to a 32 bit word.

But there's much more to this pocket-size powerhouse. It can do base conversions from binary/octal/decimal/hexadecimal modes and can store in its memory numbers in any base. It also has Shift, Rotate, Arithmetic Shift and Boolean functions that include AND, OR, XOR and NOT.

Perhaps what is most extraordinary about the CM-100 though, is not how much it can do, but how little it costs to do it. The CM-100 is the only calculator that'll let you do all your software figuring for less than you'd figure to pay for an average ($25.00) textbook.

The more you work with computers—whatever their size—the more you need a CM-100. Whether you're a student or professional, it's the one piece of hardware that will make designing your software easier.

Apple and IBM are trademarks of the Apple and IBM Corporations.

CASIO®
Where miracles never cease

Casio, Inc. Consumer Products Division: 15 Gardner Road, Fairfield, NJ 07006 (201) 882-1493, Los Angeles (213) 803-3411

CIRCLE NO. 112 ON READER SERVICE CARD

One noticeable difference of Advantage is its *constructor linker* step. Advantage C++ generates a reference to a special function when static or external objects are declared; constructor linker searches the map file created by **link** for this function's reference, knowing that external or static objects follow it. It then splices the appropriate calls into the start-up code so that the objects are initialized correctly.

With an automatic variable allocated on the stack, calling a constructor presents no problem because its scope begins after **main()** has started. Statics and externals, however, begin their scope before **main()** has started. The same is true of destructors; for example, memory has to be freed after **main()** has finished, when the program has exited. The constructor linker patches the start-up code for **main()** in the .EXE file to correctly call constructors for any static or external objects in the program. On UNIX systems, the constructor linker also updates member-function cross references and performs other post processing.

In contrast, Guidelines imbeds a "magic number" before external or static objects. It then searches for this number in the .EXE file when the module is started, knowing an object to be constructed should follow this number. This essentially defers the constructor linking until runtime.

Although this method eliminates the extra constructor-linker step, it can cause serious problems if static code or data in the program happen to match the magic number. Given the fact that C++ takes such immense care to ensure code reliability, the magic-number method seems rather unreliable.

Table 1 shows the basic specifications of these two packages, which are quite similar. They both support the language as defined in the Stroustrup book and require fairly hefty resources, including a hard disk if the Microsoft compiler and libraries are included.

TABLE 1: *Compiler Specifications*

	GUIDELINES	LIFEBOAT
Product	Guidelines C++	Advantage C++
Version tested	1.1	1.1M3
Price	$195	$495
Minimum disk space required[a]	720KB	720KB
Minimum RAM	512KB	640KB
Supports full C++ language	●	●
Supports Microsoft C extensions	○	●
Memory models supported		
Large	●	●
Medium	●	○
Compact	●	○
Small	●	●

[a] *Does not include Microsoft C compiler and libraries.*
● = *Yes*
○ = *No*

Their common heritage in the AT&T C++ compiler makes these two products basically similar; however, some important differences must be considered.

GUIDELINES GAPS

Guidelines C++ is installed using a batch file supplied with the package. The batch file builds a subdirectory called \CPP for the software, then allocates subdirectories named \CPP\BIN, \CPP\LIB, and \CPP\INCLUDE, mirroring the organization of Microsoft C. The BIN and LIB environment variables must be set to recognize these new directories for compilation.

The Guidelines C++ documentation is rather sketchy; its manual consists of a very slim section on installing Guidelines C++, some UNIX-format reference pages on the actual programs, and AT&T's C++ Translator release notes. In addition, the documentation that is provided is quite poor: it has no index, the pages give the appearance of being single-sided photocopies, and it contains barely enough information to install and use the software.

Similar to Advantage, Guidelines replaces the Microsoft compiler driver with one that processes files through the C++ translator. No constructor linker is required for Guidelines C++, saving an extra preparation step. The driver is one of several batch files (one for each model type and compilation target) that can be modified easily.

The Guidelines preprocessor does not support the range of options provided by the Advantage preprocessor (discussed below), making it somewhat easier to use. However, the Advantage compiler can be used as a front-end to both Microsoft C and C++, but the Guidelines processor can compile only C++ programs. It has no provision for compiling C-only programs and passing switches to the Microsoft compiler.

ADVANTAGE IDIOSYNCRASIES

Installation of Advantage C++ is relatively straightforward. A batch file is provided to copy the preprocessor, libraries, header files, and sample programs into appropriate directories.

Only passing mention is made of the fact that the Microsoft large-model libraries must be present for the preprocessor to operate. In fact, Advantage supports only the large and small models, which could cause problems in applications using other memory models. In addition, although the documentation lists 512KB as the minimum amount of RAM needed, even the smallest sample program would not compile correctly until 640KB was installed on a DOS 3.2 system.

Advantage C++ includes on diskettes all the programs in *The C++ Programming Language*, which makes learning C++ by self study easier. However, the documentation lacks an index and is poorly organized. The manual consists of the Stroustrup book, an additional 58 pages describing the C++ implementation for both the Lattice and Microsoft compilers, and the AT&T C++ translator documentation.

In normal use, Advantage C++ acts like a complete driver for the compilation process. It compiles groups of C++ or regular C programs with a single command. Prefix characters segregate compilation switches for use by either the C++ or the C compiler.

After compilation, **link** generates a .EXE file, followed by a constructor linker. The Advantage driver does not handle the Plink linker by Phoenix Computer Products, thus disallowing that linker's overlay facility, which is useful in large programs.

Advantage's development environment is very similar to the standard Microsoft C environment, because the additional .EXE files, libraries, and header files are all installed into the appropriate Microsoft directories. Other than the added compilation steps (and some changes to Microsoft compilation switches), using Advantage C++ is a relatively painless, but slow, process.

TABLE 2: *Benchmark Results*

	PREP	COMPILE	LINK	CONSTRUCT	TOTAL	RUN
SIEVE						
MSC only	N/A	27.9	16.0	N/A	43.9	87.3
Guidelines	21.0	25.3	16.2	N/A	62.5	87.2
Lifeboat	28.3	27.1	24.2	2.8	82.4	87.3
INTSET						
Guidelines	50.2	58.0	31.4	N/A	139.6	4.7
Lifeboat	62.3	59.2	41.7	15.8	179.0	5.1
COMP						
Guidelines	61.0	46.4	38.8	N/A	146.2	1.4
Lifeboat	83.3	48.2	46.7	19.3	197.5	1.9
NODES						
Guidelines	23.9	63.3	16.3	N/A	103.5	0.9
Lifeboat	23.0	62.4	24.0	3.0	112.4	1.1

Times shown are in seconds.
Benchmarks were done on an IBM PC/XT with 640KB memory, using DOS 3.2.

C++ was designed to add very little execution overhead to programs, and it succeeds in this respect. These two products are implemented as preprocessors, which extends the already-lengthy compile times of the Microsoft C compiler (version 4.0).

MAKING THE C++ GRADE

Four sample test programs were compiled and run by both compilers: **sieve**, **intset**, **nodes**, and **comp**. The Eratosthenes Sieve program was run through the Microsoft C compiler alone to have an indication of the additional overhead generated in C++ programs. **Intset**, taken from the Stroustrup book, generates a set of integers, then prints its members. **Nodes**, also from the book, tests compilation of virtual functions and class inheritance. **Comp** defines a class of complex numbers and creates two objects from it. The Guidelines compiler, as shipped, could not run this test because a function was missing from its library. A call and a quick patch from Guidelines Software resolved the problem.

The test machine was a PC/XT with a 10MB hard disk and 640KB of memory running DOS 3.2. CONFIG.SYS had BUFFERS = 20 and FILES = 20 and the system was rebooted between tests to eliminate the effect of disk buffering. Test results are shown in table 2. The constructor-linker step was timed separately, and this step was included only for the Advantage compiler.

Overall, Guidelines C++ tested slightly faster than the Advantage C++. The Advantage constructor-linker step also added a few seconds to the compile times; however, both compilers added appreciable time to what is already a slow compilation process with the Microsoft C compiler.

The object modules generated by the two compilers also were very close in performance, and close to the baseline performance of native C code. Neither C++ compiler added much runtime overhead to normal C code, as shown in the Sieve timings.

The tests confirm that, although programs take longer to process through the C++ compilers than through a C-only compiler, the code's performance levels are close to standard C code. Thus, C++ could be an effective tool where the slower compilation times are not as appreciable, such as in a 80386-based system.

Both Advantage C++ and Guidelines C++ are essentially ports of the AT&T C++ compiler running under UNIX System V. Their size and speed are comparable, although Guidelines is slightly faster. The Advantage documentation is better, but the Advantage package is also much more expensive. Both suffer from sketchy documentation that is well below the current standard for C compilers on the PC.

Unlike other object-oriented programming languages, such as Actor and Smalltalk-80, C++ is designed to work on top of an existing low-level language. But this construction is a double-edged sword. The tests show the performance of compiled C++ programs is extremely good because of the optimization available from the Microsoft compiler; however, the driver syntax and environment required to use C++ is cumbersome and the extra processing time is a nuisance.

Although C++ is a better C language, incorporating many improvements over standard C, its implementation as a preprocessor could hurt programmer productivity by adding extra time to the process of compiling and linking programs. Because the fast Borland Turbo C and Quick C compilers finally are becoming available, C++ is an anachronism in many ways—an attempt to improve programmer productivity by adding new statements to C, rather than improving the process of writing and debugging programs in C.

For programmers developing sizeable software packages with a large team of programmers, the extra expressive power and syntactic rigor of C++ is well worth examining closer. For the rest, C++ is not recommended until a faster programming environment can be provided.

Guidelines Software
P.O. Box 749
Orinda, CA 94563
415/254-9393
Guidelines C++ 1.1
CIRCLE 354 ON READER SERVICE CARD

Lifeboat Associates
55 South Broadway
Tarrytown, NY 10591
800/847-7078; 914/332-1875
Advantage C++ 1.1M3
CIRCLE 355 ON READER SERVICE CARD

Microsoft Corporation
16011 N.E. 36th Way
P.O. Box 97017
Redmond, WA 98073
800/426-9400; 206/882-8080
Microsoft C Compiler 4.0
CIRCLE 356 ON READER SERVICE CARD

Marty Franz is a programmer for Allen Test Products, a division of The Allen Group, Inc., located in Kalamazoo, Michigan.

LISTING 1: INTSET.CXX

```
#include <stream.hxx>

class intset {
      int cursize, maxsize;
      int *x;
public:
      intset(int m, int n);
      ~intset();

      int member(int t);
      void insert(int t);

      void iterate(int& i)  { i = 0; }
      int ok(int& i)        { return i<cursize; }
      int next(int& i)      { return x[i++]; }
};

extern void exit (int);

void error(char *s)
{
     cout << "set: " << s << "\n";
     exit(1);
}

extern int atoi(char *);

extern int rand();

int randint (int u)  // in the range 1..u
{
     int r = rand();
     if (r < 0) r = -r;
     return 1 + r%u ;
}

intset::intset(int m, int n)
{
     if (m<1 || n<m) error("illegal intset size");
     cursize = 0;
     maxsize = m;
     x = new int[maxsize];
}

intset::~intset()
{
    delete x;
}

void intset::insert(int t)
{
     if (++cursize > maxsize) error("too many elements");
     int i = cursize-1;
     x[i] = t;

     while (i>0 && x[i-1]>x[i]) {
         int t = x[i];
         x[i] = x[i-1];
         x[i-1] = t;
         i--;
     }
}

int intset::member(int t)
{
     int l = 0;
     int u = cursize-1;

     int m =0;
     while (l <= u) {
         m = (l+u)/2;
         if (t < x[m])
            u = m-1;
         else if (t > x[m])
            l = m+1;
         else
            return 1;    // found
     }
     return 0;    // not found
}

void print_in_order(intset* set)
{
   int var;
   set->iterate(var);
   while (set->ok(var)) cout << set->next(var) << "\n";
}

main ()
{
   int count = 0;
   int m = 100;
   int n = 10000;
   intset s(m,n);

   int t = 0;
   while (count <m) {
       t = randint(n);
       if (s.member(t)==0) {
          s.insert(t);
          count++;
       }
   }
   print_in_order(&s);
}
```

LISTING 2: INTSET.C

```
#line 1 "intset.cpp"

/* <<cfront 05/20/86>> */
/* < intset.cpp */

#line 1 "intset.cpp"

#line 4 "intset.cpp"
struct intset { /* sizeof = 6 */
int _intset_cursize ;

#line 5 "intset.cpp"
int _intset_maxsize ;
int *_intset_x ;
} ;
struct intset *_intset__ctor ();
int _intset__dtor ();

#line 11 "intset.cpp"
int _intset_member ();
int _intset_insert ();

#line 19 "intset.cpp"
extern int exit ();

#line 21 "intset.cpp"
int error (_auto_s )char *_auto_s ;
{
#line 23 "intset.cpp"
_ostream__lshiftFPC__ ( (struct ostream *)_ostream__lshiftFPC__ (
(struct ostream *)_ostream__lshiftFPC__ ( & cout , (char *)"set:
") , (char *)_auto_s ) ,
#line 23 "intset.cpp"
(char *)"\n") ;
__cPP_exit_( 1 ) ;
}
;
extern int atoi ();

#line 29 "intset.cpp"
extern int rand ();

#line 31 "intset.cpp"
int randint (_auto_u )int _auto_u ;
{
#line 33 "intset.cpp"
int _auto_r ;

#line 33 "intset.cpp"
_auto_r = rand ( ) ;
if (_auto_r < 0 )_auto_r = (- _auto_r );
```

CONNECTIVITY 386 GRAPHICS

LANS CLONES

OS-2 MICRO

MAINFRAME

TOKENRING

NETWORK PS/2

MINI

DESKTOP PUBLISHING

FACE THE FACTS AND THE FUTURE.

The industry standards are changing...new operating systems have been introduced, applications are being built with more powerful tools, and desktop work stations will be at the heart of increasingly complex development operations with a need to link both like and unlike machines and software.

How will these changes affect your computer buying, development and integration? What may have been wise hardware and software decisions a few months ago may be dangerous next month. And your state-of-the-art system may be obsolete by the end of the year.

Tune into the new technology with PC TECH JOURNAL, the magazine written for systems professionals who must tackle important development and integration decisions for their company.

No other magazine brings user, developer and integrator concerns to the table quite like PC TECH JOURNAL. We face the issues. We analyze new products to help you make wise purchasing and planning decisions. We help keep your system at optimum levels of performance.

This is the magazine written for designers, DP/MIS professionals, integrators and consultants who are involved in advanced IBM PC applications and development. We cover today's technology and tomorrow's innovations. Subscribe today, you'll receive 13 issues a year, including the special PC TECH JOURNAL DIRECTORY issue and SAVE 50% OFF the single-copy price of $53.35. Mail the attached order card today or call toll-free 1-800-852-5200.

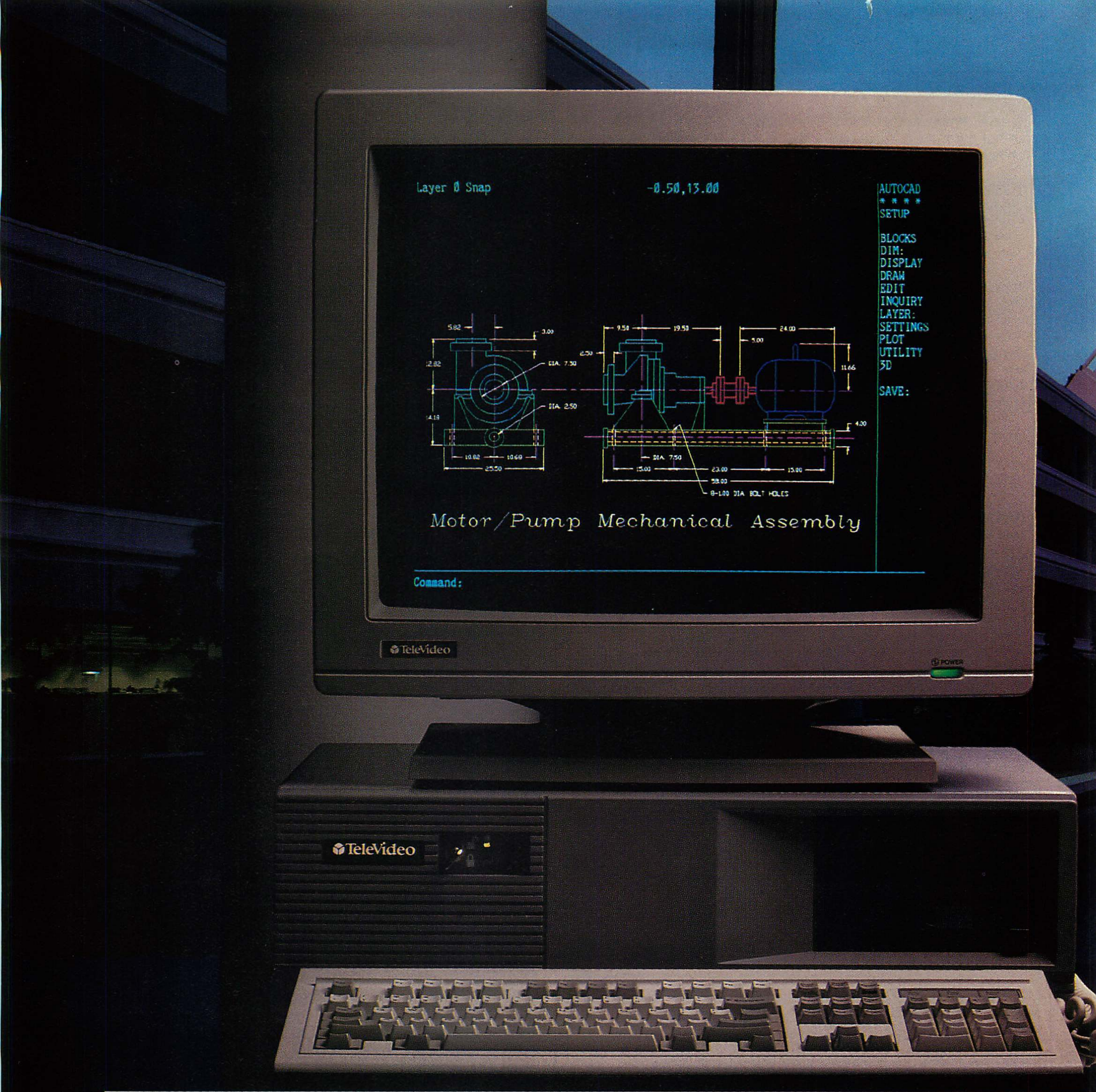
- Two years (26 issues) only $53.35. SAVE 50%!
- One year (13 issues) only $26.70. SAVE 50%!
  Basic annual subscription price $34.97.

PC TECH JOURNAL.

4Z791

```
return (1 + (_auto_r % _auto_u ));
}
;
struct intset *_intset__ctor (_auto_this , _auto_m , _auto_n )
#line 17 "intset.cpp"
register struct intset *_auto_this ;

#line 38 "intset.cpp"
int _auto_m ;

#line 38 "intset.cpp"
int _auto_n ;
{ if (_auto_this == 0 )_auto_this = (struct intset *)_new ( (long)6);
if ((_auto_m < 1 )|| (_auto_n < _auto_m ))error (
 "illegal intset size") ;
_auto_this -> _intset_cursize = 0 ;
_auto_this -> _intset_maxsize = _auto_m ;
_auto_this -> _intset_x = (((int *)_new ( 2* _auto_this ->
_intset_maxsize ) ));
return _auto_this ;
}
;

#line 46 "intset.cpp"
int _intset__dtor (_auto_this , _auto__free )
#line 17 "intset.cpp"
register struct intset *_auto_this ;
#line 49 "intset.cpp"
int _auto__free ;

#line 47 "intset.cpp"
{ if (_auto_this ){
#line 48 "intset.cpp"
_delete ( (int *)_auto_this -> _intset_x ) ;
if (_auto_this )if (_auto__free )_delete ( (int *)_auto_this ) ;
} }
;

#line 51 "intset.cpp"
int _intset_insert (_auto_this , _auto_t )
#line 17 "intset.cpp"
register struct intset *_auto_this ;

#line 51 "intset.cpp"
int _auto_t ;
{
#line 54 "intset.cpp"
int _auto_i ;

#line 53 "intset.cpp"
if ((++ _auto_this -> _intset_cursize )> _auto_this ->
_intset_maxsize )error ( "too many elements") ;
_auto_i = (_auto_this -> _intset_cursize - 1 );
(_auto_this -> _intset_x [_auto_i ])= _auto_t ;
#line 57 "intset.cpp"
while ((_auto_i > 0 )&& ((_auto_this -> _intset_x [_auto_i - 1
])> (_auto_this -> _intset_x [_auto_i ]))){
#line 58 "intset.cpp"
int _auto_t ;

#line 58 "intset.cpp"
_auto_t = (_auto_this -> _intset_x [_auto_i ]);
(_auto_this -> _intset_x [_auto_i ])= (_auto_this -> _intset_x
[_auto_i - 1 ]);
(_auto_this -> _intset_x [_auto_i - 1 ])= _auto_t ;
_auto_i -- ;
}
}
;
int _intset_member (_auto_this , _auto_t )
#line 17 "intset.cpp"
register struct intset *_auto_this ;

#line 65 "intset.cpp"
int _auto_t ;
{
#line 67 "intset.cpp"
int _auto_l ;
int _auto_u ;

#line 70 "intset.cpp"
```

```
int _auto_m ;

#line 67 "intset.cpp"
_auto_l = 0 ;
_auto_u = (_auto_this -> _intset_cursize - 1 );

#line 70 "intset.cpp"
_auto_m = 0 ;
while (_auto_l <= _auto_u ){
#line 72 "intset.cpp"
_auto_m = ((_auto_l + _auto_u )/ 2 );
if (_auto_t < (_auto_this -> _intset_x [_auto_m ]))
#line 74 "intset.cpp"
_auto_u = (_auto_m - 1 );
else if (_auto_t > (_auto_this -> _intset_x [_auto_m ]))
#line 76 "intset.cpp"
_auto_l = (_auto_m + 1 );
else
#line 78 "intset.cpp"
return 1 ;
}
return (int )0 ;
}
;
int print_in_order (_auto_set )struct intset *_auto_set ;
{
#line 85 "intset.cpp"
int _auto_var ;

#line 86 "intset.cpp"
int _auto__Xa__lshiftFI___ostream ;
( (*((int *)(& _auto_var )))= 0 ) ;

#line 87 "intset.cpp"
while (( (*((int *)(& _auto_var )))< ((struct intset *)_auto_set
)-> _intset_cursize ) )_ostream__lshiftFPC__ ( (struct ostream
*)( (_auto__Xa__lshiftFI___ostream = ( ((struct
#line 87 "intset.cpp"
intset *)_auto_set )-> _intset_x [(*((int *)(& _auto_var )))++ ])
), ( _ostream__lshiftFL__ ( ((struct ostream *)(& cout )), ((long
)_auto__Xa__lshiftFI___ostream )) )
#line 87 "intset.cpp"
) , (char *)"\n") ;
};
int main (){ _main();
#line 91 "intset.cpp"
{
#line 92 "intset.cpp"
int _auto_count ;
int _auto_m ;
int _auto_n ;
struct intset _auto_s ;

#line 97 "intset.cpp"
int _auto_t ;

#line 92 "intset.cpp"
_auto_count = 0 ;
_auto_m = 100 ;
_auto_n = 10000 ;
_intset__ctor ( & _auto_s , _auto_m , _auto_n ) ;

#line 97 "intset.cpp"
_auto_t = 0 ;
while (_auto_count < _auto_m ){
#line 99 "intset.cpp"
_auto_t = randint ( _auto_n ) ;
if (_intset_member ( & _auto_s , _auto_t ) == 0 ){
#line 101 "intset.cpp"
_intset_insert ( & _auto_s , _auto_t ) ;
_auto_count ++ ;
}
}
print_in_order ( & _auto_s ) ;

#line 95 "intset.cpp"
_intset__dtor ( & _auto_s , (int )0 ) ;
}
__cPP_exit_(0);};
/* the end */
```



Now The Only Limit Is Your Imagination.

Introducing TeleSTAR, the first open system 386 engineering workstation.

The new TeleVideo® TeleSTAR/386™ Engineering Workstation is the first workstation to give you 16 MHz 80386 power, high resolution graphics and Ethernet™ networking. It even comes with an affordable price: just $10,995.

The new TeleSTAR/386 is an "open system," with all the features you expect to find on a Motorola 68020-based workstation. It's the only way to get all the flexibility, portability and connectivity you need.

It uses Microport™ DOSMerge 386™ for concurrent UNIX® system V.3 and MS-DOS™ operation to give you access to the widest range of technical and business software. It also features MIT's X Window,™ the de facto standard user-interface system for multitasking on one screen, plus GKS software for stan-

dard graphics applications.

To present your work at its best, there's a 1280 x 1024 pixel high-resolution display and 256 simultaneous colors from a palette of 4,096 (with 16.7 million colors optional), and a monochrome display with the same high-resolution and four grey levels. There's even a mathematical co-processor to help speed up your calculations.

And to help you share the wealth, there's a built-in Ethernet interface with TCP/IP and RFS software under UNIX system V.3 supporting NFS™ from Sun Microsystems. Together, they let you network heterogeneously with mainframes, stand-alone workstations and PCs.

Our TeleSTAR/386 Engineering Workstation comes with features you might not expect at such a low price. Including a 17-inch color monitor (a 19-inch monochrome monitor is optional). A 3-button optical mouse. And 4 Mbytes of 32-bit RAM memory that's expandable to 16 Mbytes.

If you're ready for a workstation that gives you the power you need at a price you can afford, you're ready for the new TeleSTAR/386.

Call us toll-free or write today, for more information.

TeleVideo Systems, Inc., 1170 Morse Avenue, Sunnyvale, CA 94088-3568.

TeleVideo®

THE VISION YOU NEED TO SUCCEED.

CIRCLE NO. 182 ON READER SERVICE CARD

Call 1-800-835-3228

Simple LAN Alternatives

Inexpensive switches—either mechanical or electronic—can be more effective than LANs for sharing peripherals.

MICHAEL HURWICZ

Local area networks are almost the automatic answer to all questions of sharing equipment. Using a LAN to share a printer or another peripheral, however, is like hitting a thumbtack with a sledge hammer.

An alternative method is to use a *peripheral-sharing switch*. The simplest of these switches lets two computers share one peripheral; a two-position knob on the switch selects which computer drives the peripheral. More sophisticated electronic switches eliminate manual switching and provide output buffering for multiple computers.

Most computers do not require any hardware or software changes to use a peripheral-sharing switch. On each computer, the serial or parallel port normally connected directly to the peripheral is instead connected to the switch; a cable leads from the switch to the peripheral that is being shared. Applications for the most part use the peripheral as if it were connected directly to the computer.

LANS VS. SWITCHES

Peripheral-sharing switches can be useful in many small businesses and departments for sharing printers, plotters, and modems; to perform occasional file transfers; or to provide a small local electronic mail system. If a LAN is already installed or added later to a switch installation, the switch can be integrated into the LAN (see figure 1). All computers on the LAN can use the printer through the LAN's print server, which is connected to the switch. Computers that are not on the LAN, perhaps because their applications cannot spare the memory a LAN requires, also can access the printer through the switch.

For these smaller installations, peripheral-sharing switches have many advantages over LANs, foremost of which may be cost. LANs usually are connected through interface cards that cost $300 or more; switches need only a serial or parallel port, which is either already available or can be purchased for about $60. The switch itself can cost as little as $25 per computer.

LANs require more memory—60KB to 100KB for network drivers, leaving less for applications; switches do not need drivers. Only a few LANs can accommodate computers having different operating systems or bus architectures. A switch, on the other hand, easily can accommodate a multitude of computers that support parallel or serial I/O, including a Macintosh, an Apple II, a PC running DOS, a PC running XENIX, and many others.

Of course, peripheral-sharing switches are more limited than LANs. Only a few switches can deliver the

PHOTOGRAPHY: MARC DAVID COHEN

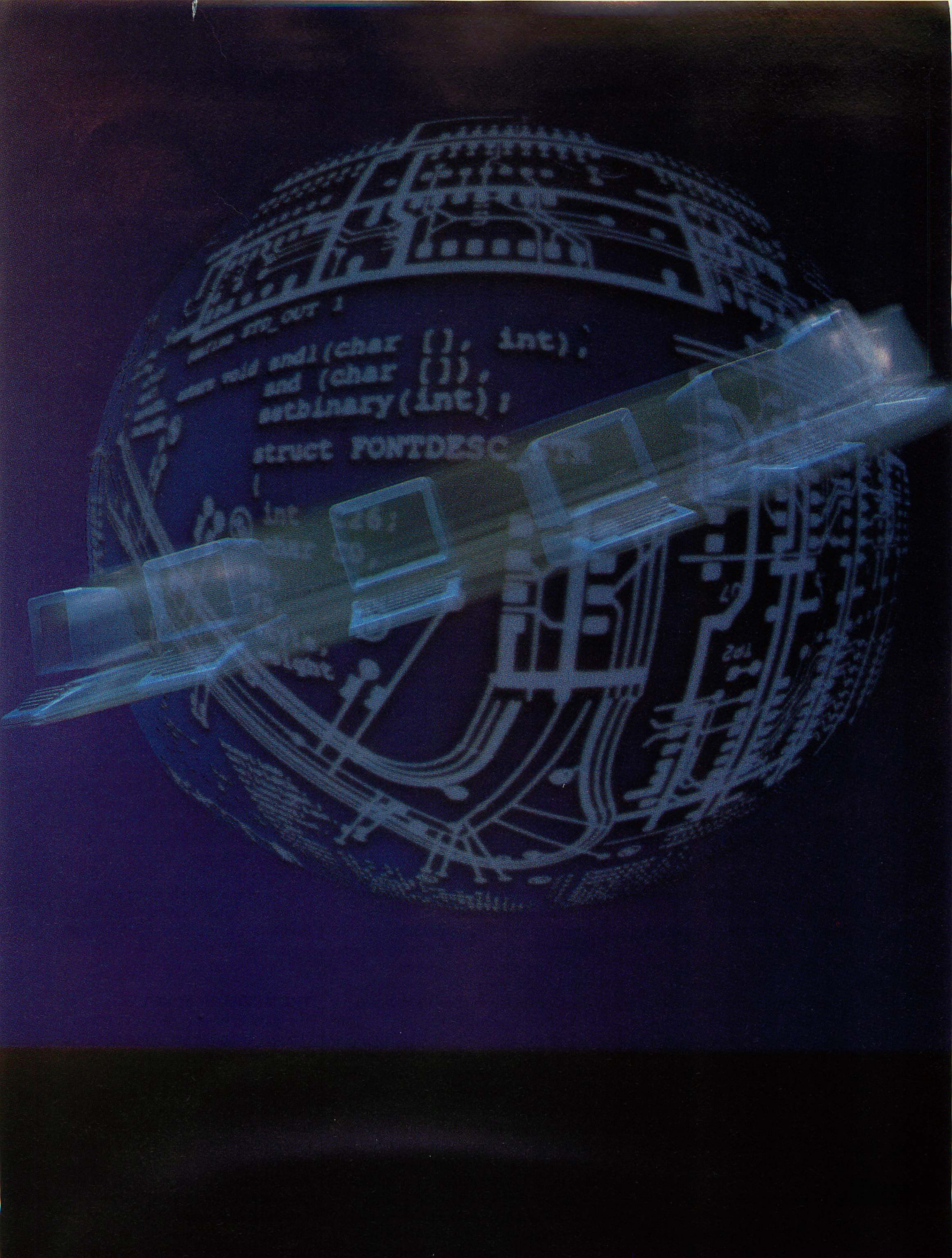
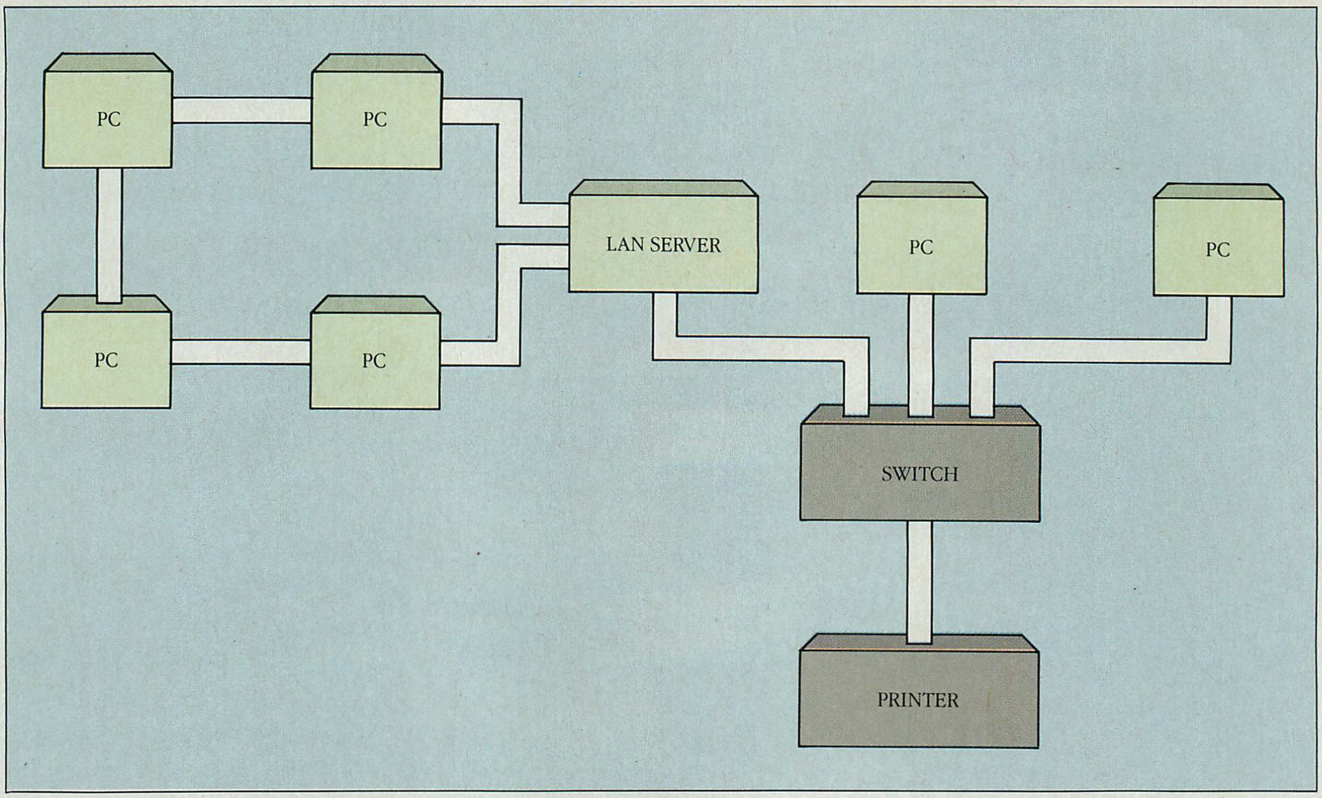

setbinary(int);
struct FONTDESC

FIGURE 1: *LANs with Peripheral-sharing Switches*



Even when a LAN is installed, a switch can be useful. Here, two non-networked PCs share the printer with the network.

more sophisticated LAN functions, such as file sharing, electronic mail and messaging, and multiuser applications. Although some serial switches allow two computers to communicate and perform file transfers, they lack the transparent operation of a LAN and are much slower; the transfer occurs at serial-port speeds (typically 300 to 19.2K baud), rather than at LAN speeds (typically 1M to 10M bits per second).

This low speed also limits growth. With a LAN, hundreds of PCs and peripherals can be built into one network; with switches, they cannot. Typical switch installations contain two to eight PCs sharing one or two printers. Most of the more sophisticated switches can handle perhaps a modem, a plotter, and a few links to mainframes or minicomputers. Adding more peripherals would push the technology farther than it was designed to go.

LANs are designed for convenient multiuser database access; switches are not. Although some switches let a user view a file on another computer, only one user can access a computer at a time. Access is through the RS-232 port, which can accommodate only one remote computer at a time. Some so-called RS-232 LANs, which use an RS-232, or asynchronous, port as LAN hardware, overcome this limitation, allowing low-speed multiuser access through the RS-232 port. However, no units tested provided this feature.

CONSIDERING SWITCHES

Peripheral-sharing switches can be either mechanical or electronic. A mechanical switch has one or more manual controls to select the computer and/or peripheral to be connected. An electronic switch detects activity on its input port when a computer transmits data and automatically connects the computer to a free output port (and its attached peripheral). If the output port is not free, the switch returns a busy signal to the computer.

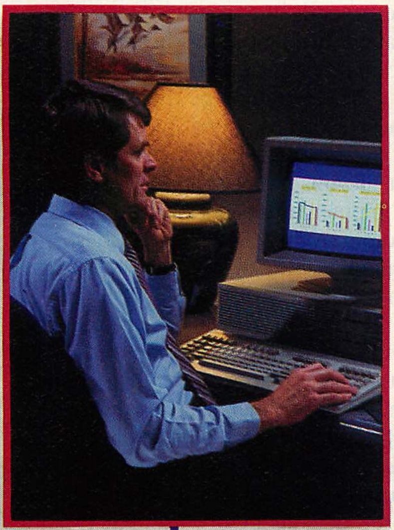
Electronic switches are either *N-to-N* or *any-to-any*. In an N-to-N switch, each port must be configured for input or output. Two-to-one switches, for example, have two inputs and one output. Other common sizes are two-to-two, three-to-one, four-to-one, and four-to-two switches. In an any-to-any switch, any serial port can connect to any other serial port for both input and output, but each parallel port still must be configured for one or the other.

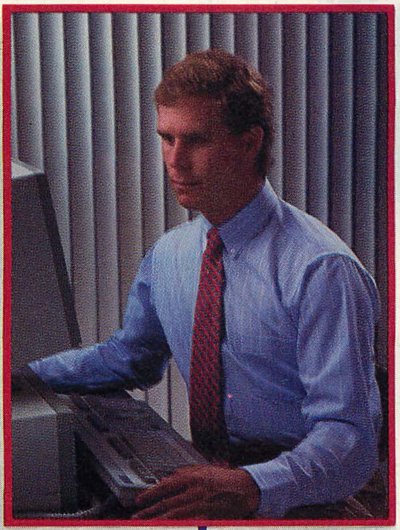
Other key considerations in evaluating switches are the presence of an internal buffer, support for multiple simultaneous inputs, and provision for interconnecting serial inputs and parallel outputs. When a switch is said to support multiple, "simultaneous" inputs, input is not truly simultaneous, although it appears to be. The switch reserves buffer space for each input port and continues to scan input ports even while taking input. If input is detected on a second port, the switch jumps quickly back and forth between the two ports, allotting some buffer space to each. Switches that do not allow simultaneous input permit data from one input port to fill the entire buffer and do not free the buffer until that port becomes inactive.

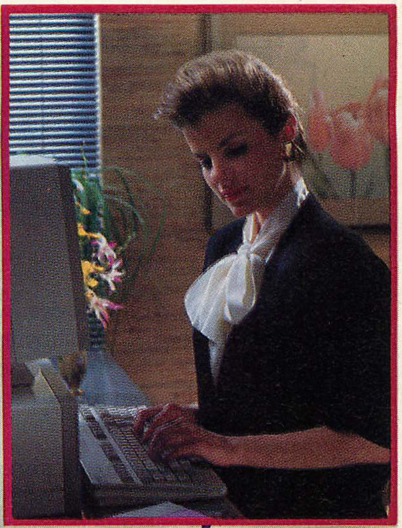
THE MECHANICAL ROUTE

The most appealing aspect of a mechanical switch is price—from $25 to $200 per computer. Such a switch is ideal for two to four PCs in the same area that need occasional access to a peripheral. A single switch seldom is used to support more than seven computers. Line extenders can increase distances and switches can be cascaded or daisy-chained to accommodate more devices (see figure 2). However, switching in a cascaded configuration is cumbersome because the user must set two switches to make the connection.

Print Master lets people share printers







Get the most out of your expensive resources—your people, your computers and your printers—with Print Master by BayTech. Not only does it let your people and computers share all of your printers, its buffer keeps them working instead of waiting.

Easy to set up, easy to use

Initial setup is menu-prompted and allows you to configure Print Master to your application. You decide which ports are input and which are output. You also set such functions as baud rates, handshaking, timeout, how you select a printer, etc.

After setup, you simply cable Print Master between your computers and printers, and you're ready to go. BayTech's Memory Resident Program lets your PC users select printers via function keys that correspond to a menu window.

Print Master keeps everything running

All users can send data simultaneously to Print Master's dynamically allocated buffer, keeping your people and computers working. This first-in-first-out buffer also sends data to all printers simultaneously to keep them running at full capacity. The buffer size? A big 512K that's expandable to one megabyte.

Eleven models with serial and parallel

An advantage of Print Master is its flexibility: Any port can be a computer port or a printer port. Choose from eleven models with different combinations of serial and parallel ports. Print Master internally converts serial to parallel and vice versa on combination serial/parallel models. Six ports, $795. Eight ports, $895. Ten ports, $995.

Non-buffered models from $339. GSA pricing available.

Want details?

Contact your dealer or Bay Technical Associates for information about Print Master and BayTech's complete line of data communications products, proudly made in the U.S.A.

BayTech

Bay Technical Associates, Inc.
Data Communications Products Division
200 N. Second St., P.O. Box 387
Bay Saint Louis, Mississippi 39520
Telex 910-333-1618 BAYTECH
Phone 601-467-8231 or
800-523-2702

Circle No. 126 for Dealer, No. 129 for End-User.

LAN ALTERNATIVES

The most common mechanical switches are two-way, three-way, four-way, and cross-matrix. The two-way switch connects two computers to one peripheral; the three-way, three computers to one peripheral; and the four-way, four computers to one peripheral. The cross-matrix switch connects two peripherals and two computers, alternating the peripherals between the computers at the turn of the switch.

The terminology used to identify these switches is not always consistent. A two-way switch is often called an AB or an ABC switch, with *A* and *B* the computer inputs and *C* the peripheral output. A three-way switch is also called an ABC or ABCD switch, and a four-way, an ABCD or ABCDE.

Mechanical switches are reliable and easy to use and install, requiring little more than simply plugging them in. Because other users share the peripheral, the main adjustment with a mechanical switch is resetting all the peripheral's features to the user's desired defaults before use. For example, sending a form feed at the beginning of each job ensures that a printer will begin printing at the top of a new page. Many applications allow the user to send a printer set-up string before each print job. Alternatively, the user can create a short program to send control codes to the printer.

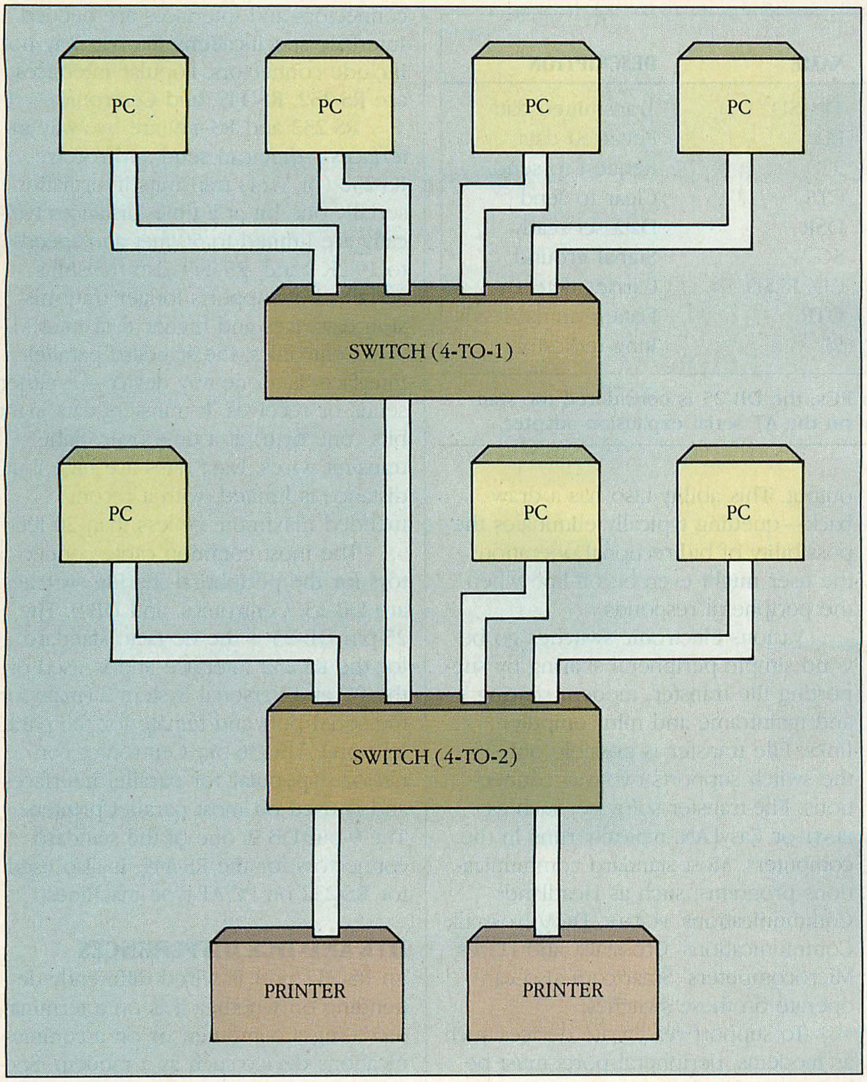
The bare-bones simplicity of the typical mechanical switch makes it more flexible than many of its electronic cousins. Mechanical switches are not affected by the load on the end of the line; and their connections are indistinguishable from cabled connections, except that they can be broken by the flip of a switch. Like direct-cable connections, the number of signaling lines supported is important; not all switches support the full 25 lines on a DB-25 connector. For the same reason, some electronic switches do not allow modems or plotters to be shared.

With mechanical switches, only a few devices can be switched, and all switching is manual, which can be a problem when the computers, peripheral devices, and/or switches are in different rooms. Changing positions on the switch also may generate a small amount of noise on the line, which can transmit to the device and result in a garbled first line of output—another good reason to insert a form feed before any output on a printer.

ELECTRONIC CHOICES

Electronic switches solve some of the problems associated with mechanical

FIGURE 2: *Daisy-chained Switches*



Larger configurations can be created by daisy-chaining multiple switches. In the configuration illustrated here, a 4-to-1 switch is fed into a 4-to-2 switch.

switches. When an electronic switch senses activity on one of its input ports, it automatically connects the computer attached to that port to an available printer (or other peripheral). After a period of inactivity on the port, the switch breaks the connection, thereby freeing the device for input from another port. Some products allow the added convenience of setting the time-out period individually for each port.

Although electronic switches have more capacity than do mechanical switches, an electronic switch that can handle 16 devices, including computers and peripherals, is considered large, although Giltronix does make the EZ-Queue 3000 switch with up to 24 serial inputs and two parallel outputs. As with mechanical switches, daisy-chaining can increase the number of ports served (see figure 2). Unlike mechanical switches, however, daisy chaining does not complicate the switching, because it is automatic. However, a service priority is implicit in daisy chaining—the more switches a computer must traverse, the lower its priority.

Electronic switches are much more likely than mechanical switches to offer special features such as buffering, extended-distance cabling, programmable control, remote programming, and multiple concurrent users. By supporting features such as these, a peripheral-sharing switch moves closer to being a general-purpose switch.

Buffering, which holds print data in a switch's internal memory, can speed PC applications by allowing a new print job to be accepted while the previous job continues to print. Although inserting a stand-alone buffer between the switch and the printer is

TABLE 1: *RS-232 Pin Descriptions*

CONNECTOR PIN NUMBERS DB-25 (XT, PS/2)	DB-9 (AT)	NAME	DESCRIPTION
2	2	TD, SD	Transmitted data
3	3	RD	Received data
4	7	RTS	Request to send
5	8	CTS	Clear to send
6	6	DSR	Data set ready
7	5	SG	Signal ground
8	1	CD, RLSD	Carrier detect
20	4	DTR	Data terminal ready
22	9	RI	Ring indicator

Two RS-232 serial connectors are used on PCs; the DB-25 is considered the standard connector; DB-9 is used by IBM only on the AT serial expansion adapter.

easy and sometimes more economical, a stand-alone buffer would not permit multiple simultaneous inputs, as buffered switches typically do.

However, buffering is a mixed blessing; it can cause the failure of direct-wired file-transfer applications, such as BLAST from Communications Research Group or EasyLAN from Server Technology. Because buffered switches hold information in memory, machine-to-machine transfer times are too long and unpredictable for these applications, which have tight timing requirements to increase efficiency.

A buffer can be confusing when used with printer queuing software. The DOS PRINT command, for example, displays the queue of files waiting to be printed, but cannot identify files already printed and still in the buffer. A command such as PRINT/T, which terminates printing, does not affect files already buffered for printing.

Some electronic switches provide extended-distance cabling by conditioning the signal internally, which permits greater distances between computers and printers. A stand-alone line driver also can be used to extend distances; sometimes called an *interface extender*, this is a signal-conditioning device and is typically housed in a small box.

Many electronic switches also can be programmed on a port-by-port basis for communications parameters, printer set-up strings, and baud rates. With mechanical switches, these parameters are typically set at each computer using applications software or special programs. Most programmable switches also offer remote programming, allowing the user to program the switch from an attached PC or terminal.

Some switches can handle input from multiple concurrent users, holding the input in buffer queues for later output. This ability also has a drawback—queuing typically eliminates the possibility of bidirectional operation; the user might even be off-line when the peripheral responds.

Various electronic switches go beyond simple peripheral sharing by supporting file transfer, modem sharing, and mainframe and minicomputer links. File transfer is possible only if the switch supports two-way connections. File-transfer software, such as BLAST or EasyLAN, typically runs in the computers. Most standard communications programs, such as Headlands Communications' PC-TALK, DCA/Crosstalk Communications' Crosstalk, and Hayes Microcomputers' Smartcom also can operate on these switches.

To support read/write devices such as modems, peripheral ports must be bidirectional. Modems also require more complicated forms of signaling than printers do. A mainframe or minicomputer can be treated as a standard read/write asynchronous device, or the switch can support or emulate particular types of terminals.

CABLE VAGARIES

Cabling is likely to cause some difficulty when working with printer-sharing boxes. The user may—or may not—be able to use the current printer-to-computer cable as a computer-to-switch or switch-to-printer cable. In fact, some switches require cable types not commonly available, such as an output (switch-to-printer) cable with 36-pin Centronics connectors on both ends. Others require custom-made input (computer-to-switch) cables. In addition, switches often impose severe distance limitations: 15 feet for parallel devices and 50 feet for serial devices. However, various interface extenders can overcome these limitations.

When cabling a switch, the user must first determine which types of connectors and interfaces are needed. Interface specifications may or may not include connectors. Popular interfaces are RS-232, RS-449, and Centronics.

RS-232 and RS-449 are two-way interfaces—they can send and receive. RS-232 (or V.24) transmits information serially one bit at a time; distances typically are limited to 50 feet and speeds to 19.2K baud. RS-449 also transmits serially, but supports longer transmission distances and higher data rates.

Centronics, the standard *parallel* interface, is a one-way device—it either sends or receives. It transfers data eight bits (one byte) at a time over eight transmit wires. Data rates are high, but distance is limited, with a recommended maximum of less than 20 feet.

The most common cable connectors for the peripheral-sharing switches are DB-25, Centronics, and DB-9. The 25-pin DB-25 is the de facto standard for the RS-232 interface and is used on the PC and Personal System/2 (male for the serial port and female for the parallel port). The 36-pin Centronics connector is popular for parallel interfaces and is used on most parallel printers. The 9-pin DB-9, one of the standard connectors for the RS-449, is also used for RS-232 on PC/AT-type machines.

DTE AND DCE DIFFERENCES

An RS-232 port is wired differently depending on whether it is on a terminal, including a computer, or on a communications device such as a modem. Specifically, *data terminal equipment* (DTE), typified here by a computer, is wired to transmit signals on pin 2 and receive signals on pin 3. For *data communications equipment* (DCE), typified by a modem, the wiring is exactly reversed; pin 2 receives and pin 3 transmits. Most printers are wired as DTE.

The RS-232 standard was designed for computer-to-modem communications. Table 1 describes the RS-232 lines and their pinouts for the standard (DB-25) and AT (DB-9) connectors. *Request to send* (RTS) is a signal from the computer instructing the modem to go into transmit mode. *Clear to send* (CTS) is a signal from the modem to the computer verifying that the modem is in transmit mode. *Data set ready* (DSR) informs the computer that a connection has been made, and the modem is ready for use. *Data terminal ready* (DTR) informs the modem that the computer is ready to communicate. *Received line-signal detector* (RLSD) (or carrier detected) is a modem-to-com-

Choose your favorite. $475*

We're not selling printers, but a new networking device that'll make it seem like you have an extra one. Or two or three.

The Buffalo® SX can actually multiply the effectiveness of laser printers, dot matrix and daisy wheel printers, modems, and plotters. Anyone can send work to any printer—even when it's busy—and go back to work immediately. So it can save you thousands of dollars on extra printers you *don't* buy.

All the right connections

The Buffalo SX is an intelligent buffer, switcher and interface converter all in one. It allows several PC users to select any output device connected to the network.

There are ten ports—five parallel and five serial. Five can handle either input or output. So you have lots of options. Seven PCs sharing three peripherals. Two PCs sharing eight peripherals. Or any combination between.

But that's only the beginning. You can link more SXs together to share a set of peripherals with *any number* of PCs (of course, when you get up into the dozens, you might need another laser).

Power windows & automatic transmission

To set up the SX, just plug it in and define the ports using the switches and software provided. Concise documentation tells you how. If you have a question, call our toll-free support line.

When someone's ready to print, they simply use the handy little menu that pops up on screen with the names you've chosen. Call your printers Martha, Chatterbox, Relic, whatever.

The SX does the rest. it handles interface conversions, making parallel and serial machines work together. Yes, it even lets IBM® and Macintosh® PCs share a laser with no fiddling. And PCs can exchange files with communications software.

Best of all, it stores print jobs in its large buffer and feeds data to the printers automatically. So people can use their computers within seconds. ("Large" means a standard 256K memory, expandable to one full megabyte.)

Free software bonus

Order now and receive $100 worth of productivity tools free:

O HyperKey™ Keyboard Enhancer—increases speed and accuracy dramatically. Instant on-off for repeat keys!

O HyperDrive™ Disk Caching—makes applications run 3-5 times as fast. Fully utilizes Above™ Board memory.

O Print Booster™ for 1-2-3®—frees your PC about four times faster when printing graphs.

All you stand to lose are long coffee breaks

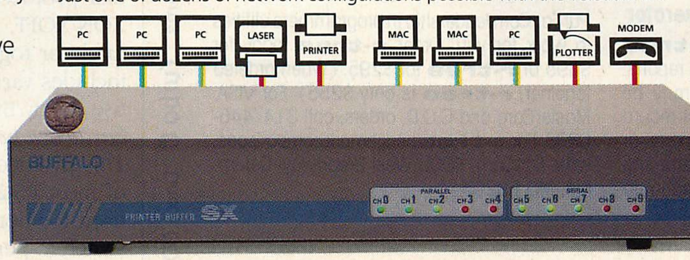
If you've tried a printer buffer, you know how limited most are. The SX is different. It's the slickest productivity booster this side of a $10,000 LAN. It's so dependable, it comes with a full one-year warranty. Plus this money-back guarantee:

Order a Buffalo SX. Try it for 45 days. If you can part with it, send it back for an immediate refund. No questions asked. And you can keep the productivity software.

We're certain you *won't* give it up once you've tried it. We think you'll add more. Ask about our other low-cost buffer/switchers too. There's one for every application imaginable.

Call now. Use Visa, MasterCard or American Express. You'll save hours of frustration. Plus the cost of another laser!

Just one of dozens of network configurations possible with the versatile SX.



Buffalo Products, 859 Cowan Road, Burlingame, CA 94010 **(800) 345-2356**; In California **(800) 543-2356**

CIRCLE 106 ON READER SERVICE CARD

BUFFALO

*This special direct introductory price will change without notice after the first 5,000 Buffalo boxes are shipped. You *must* mention this code number to qualify: PCT9

puter signal indicating that the local modem is receiving an appropriate signal from the remote modem.

When a DCE port is connected to a DTE port, a straight-through cable can connect the identically numbered pins of each connector attached to each end of the cable. When two like ports are connected (DTE to DTE or DCE to DCE), a *null-modem cable*, or *cross cable*, must be used; this is normally the case when connecting a computer to a printer. A null-modem cable usually crosses not only pins 2 and 3, but also pins 4 and 5 (RTS and CTS) and pins 6 and 20 (DSR and DTR), as shown in figure 3.

One common problem in setting up switches is using the wrong cable for the DTE/DCE configuration. To complicate matters, manufacturers do not always conform to the RS-232 standard when wiring the RS-232 ports. However, most manufacturers do document their use of the connector.

Mechanical switches act as a straight cable; their ports need not be configured as DTE or DCE, but the cables must be compatible with the device on the other end. One common mistake is to connect the computer to the switch with a cross cable, and then also use a cross cable to connect the switch to the peripheral. The net result is a straight-through connection.

Ports on some electronic switches are fixed as DCE or DTE; on others they can be configured. They must either conform to the existing cable or the cable must be replaced or rewired to conform to the ports.

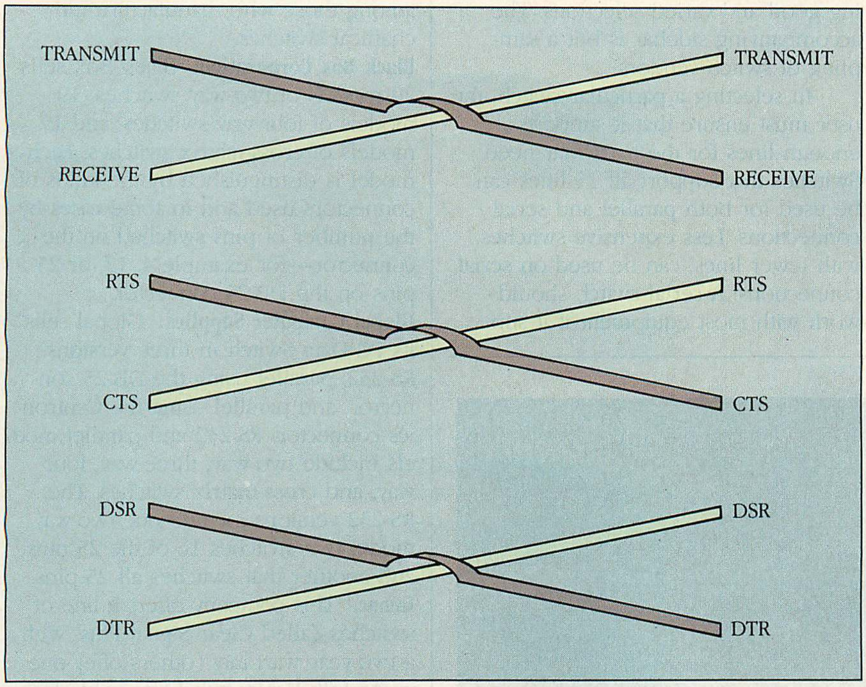
CONTROLLING THE FLOW

Flow control, or handshaking, prevents one device from sending data faster than another device can handle the data. Parallel interfaces typically use only one kind of flow control. If the connectors are correct, the parallel interface probably will work.

For RS-232 interfaces, the two popular protocols are Ready/Busy and XON/XOFF, which can be used separately or together. Each protocol also includes variants. Almost all printers use Ready/Busy exclusively, also called *hardware handshaking*, or RTS/CTS. This protocol is built into the hardware of most RS-232 interfaces and can be used for both computer-to-switch and switch-to-peripheral flow control.

In the computer-to-switch control, the switch monitors the RTS pins on its input ports, each of which is connected to an RTS pin on the computer port. The computer raises the voltage on the

FIGURE 3: *Null-modem Cable*



When two data terminal equipment (DTE) units are connected, a null-modem cable allows them to communicate by transposing the data and handshake lines.

RTS when it wants to send, and then sends the data when the switch connects and raises the CTS. If the switch does not raise the CTS, the computer cannot send. When the computer drops the RTS, the switch begins its time-out period. If this period lasts long enough, the switch breaks the connection.

In the switch-to-peripheral control, the peripheral raises the voltage on the CTS, which tells the switch it can start sending. When the voltage on the CTS falls, the switch is unable to send. In some equipment, DSR and DTR are combined with CTS.

XON/XOFF, also called *software flow control*, typically is used with graphics peripherals, such as plotters, and with communications programs and modems. Like Ready/Busy, XON/XOFF can control the flow from both the computer to the switch and switch to the peripheral. At any time, either side of the connection can send an XON character requesting more data or an XOFF character stopping the data flow. XON is DC1, or ASCII code 17; XOFF is DC2, or ASCII code 18.

Although mechanical switches are transparent to flow control, the application and the peripheral must use the same kind of flow control. With electronic switches, the switch's input-port flow control must agree with the application, and the output-port flow must agree with the peripheral. The input port usually is reconfigured dynamically by sending a configuration string to the switch from the input port.

For a serial printer, the user must set the computer and the printer for the baud rate (1,200, 2,400, 4,800, or 9,600 baud), character length (seven- or eight-bit data word), stop bits (one or two), and parity (none, odd, or even). Mechanical switches are transparent to these parameters, but electronic switches may have to be configured port by port, unless the user is satisfied with the existing defaults. Some switches can adjust automatically to any baud rate on an input port. The user still must set the baud rates for the output ports, as well as other parameters on the input and output ports.

SWITCHING SOFTWARE

Mechanical switches require little or no new software in the PC, except for resetting printer defaults and sending form feeds. Electronic switches, with their more extensive configuration requirements, come with a configuration program that allows the user to set communications parameters, DCE/DTE, baud rates, and flow-control types.

For control-code sequences that are sent repetitively to a switch or printer, the user can create files containing those sequences and then "print" them at appropriate times. For switches that support two-way commu-

nications (PC-to-PC or PC-to-modem), the user can invoke file-transfer utilities, communications programs, and RS-232 LAN software. File-transfer utilities, such as BLAST, work between PCs and other types of computers. Communications programs, such as Crosstalk and Smartcom, allow PCs to emulate various types of terminals.

MECHANICAL VENDORS

The simplicity of mechanical switches makes it difficult for the many switch manufacturers to distinguish the merits of their individual products. Mail-order houses add to the confusion by providing good and varied selections. The accompanying sidebar is but a sampling of switch vendors.

In selecting a particular switch, the user must ensure that it supports enough lines for the particular need. Switches that support all 25 lines can be used for both parallel and serial connections. Less expensive switches, with fewer lines, can be used on serial connections. A serial switch should work with most equipment if it supports pins 1 through 8 and pins 20 and 22. The following three companies are among those who manufacture mechanical switches.

Black Box Corporation. Black Box sells 20 models of two-way switches, 19 models of four-way switches, and 19 models of cross-matrix switches. Each model is distinguished by the kinds of connectors used and in some cases by the number of pins switched on the connector—for example 4, 12, or 25 pins on the DB-25 connector.

Global Computer Supplies. Global sells its PC Data Switch in three versions: RS-232, parallel using the DB-25 connector, and parallel using the Centronics connector. RS-232 and parallel models include two-way, three-way, four-way, and cross-matrix switches. The RS-232 versions include one two-way model that switches 16 of the 25 pins, and another that switches all 25 pins.

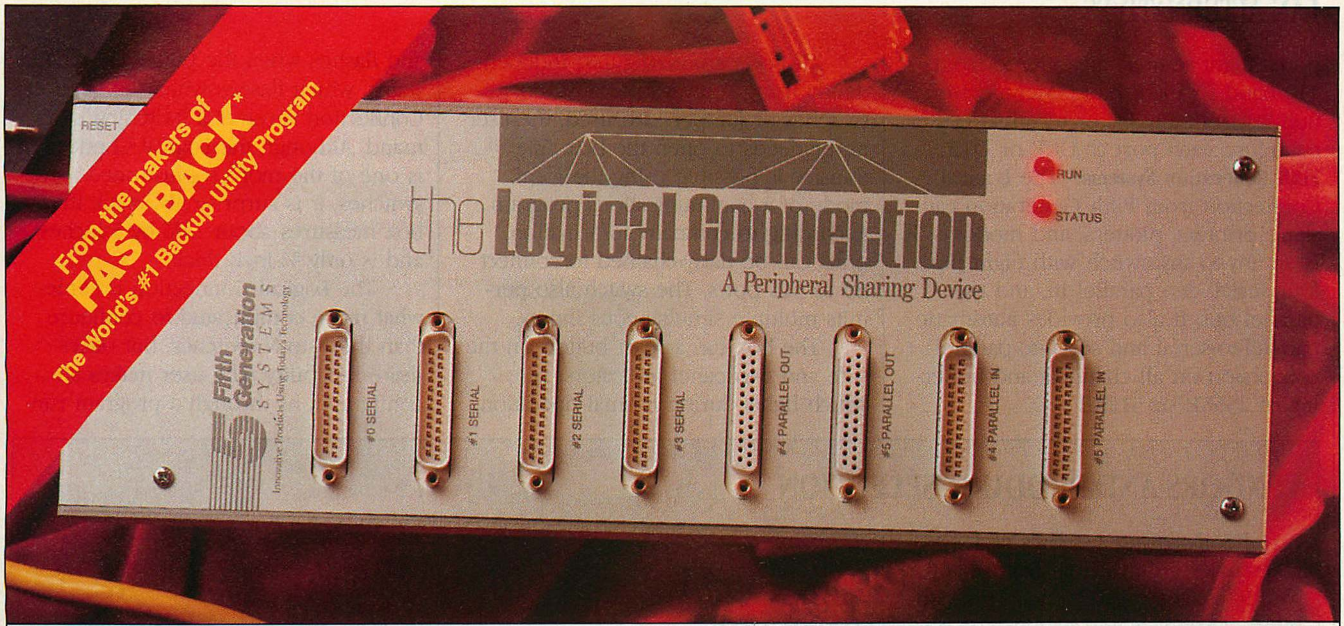
Inmac. This company offers a line of switches called Clear Signal Plus. With a two-year warranty (others offer one year), full 25-pin switching, gold electrical contacts, and an optional lock, the Clear Signal Plus line targets the high end of this low-end market.

ELECTRONIC TESTING

Although not as plentiful as mechanical switches, electronic switches can be obtained from a variety of vendors and mail-order houses (see sidebar). Buffering and switching capabilities add complexity to their design and operation. A sampling of commonly available electronic switches was tested for this article; virtually all problems that were encountered during this testing were caused by incorrectly wired cables, underscoring the importance of checking the configuration.

Digital Products. The NetCommander is an any-to-any switch offering not only printer sharing, but file transfer, local electronic mail, and data collection. It typically uses only RS-232 inputs; outputs can be serial or parallel. Models range from 4 to 16 ports. Serial-to-parallel conversion is automatic. Each output port is assigned a unique name; ports are selected by sending a control character string to the switch.

File transfer is accomplished by disabling buffering for a given input. The buffering can be disabled automatically using batch files, which Digital Products' auto-install program creates for EasyLAN. The NetCommander also can be daisy-chained. For setting baud rates on the 16-port model, ports are grouped into three groups of four, one group of two, and two independent



Your logical choice for printer & data sharing.

It's your choice.
Now you can give every PC user in your office *freedom of choice:* the choice to send their output to any printer — serial or parallel — anytime. And all you need is *The Logical Connection*™ Version 3.0.

Instant switching.
Just "pop-up" the *memory resident* switching menu, cursor-point to the device you want to switch to, and press *ENTER*.
The Logical Connection handles all the print spooling, protocols and parallel/serial conversion automatically.

Preset configurations.
10 Preset configurations give multiple computers unlimited "switch and share" access to printers, plotters and modems — and each other.
Just cursor-point to a graphic "picture" of the configuration you want and download it with a couple of keystrokes, using any IBM-PC™ or compatible. If you need a *custom* configuration, you can easily modify one of the *presets* — or create your own.

Why network if you don't need to?
For peripheral sharing applications *The Logical Connection* can outperform a LAN, right out of the box — at a far lower cost.
The Logical Connection lets minicomputers and mainframes share peripherals with PCs. Or, let many devices share a single expensive modem connection.
And for really *big* applications, you can "daisy-chain" up to 45 *Logical Connection* boxes together, up to 3/4 of a mile apart.

PC MAGAZINE EDITOR'S CHOICE

In April, 1987 *PC Magazine* named **The Logical Connection** "Editor's Choice among *all* printer sharing devices reviewed." Author Winn L. Rosch called it ". . . more than logical — maybe inspired!"

An easy choice.
The Logical Connection is easy to set up and easy to use. Best of all, it's easy on the pocketbook. For price performance and ease of use, nothing else even comes close.
So if you're looking for the *best* way to share your valuable computers and peripherals, there's only one logical choice. Get the logical connection today.
For the name of your nearest dealer, or to order direct, call *Fifth Generation Systems at* **1-800-225-2775.**

MODEM

$495
Suggested retail price
(256K Model)

Fifth Generation
SYSTEMS, INC.
Innovative Products Using Today's Technology
11200 Industriplex Blvd., Baton Rouge, LA 70809
SALES:
2691 Richter Ave., Suite 107, Irvine, CA 92714
(800) 225-2775 • (714) 553-0111

*The Logical Connection is a trademark of Fifth Generation Systems, Inc. • IBM-PC is a trademark of International Business Machines, Inc.
*This product is in no way associated with or has originated in Fisher Scientific or Allied Corporation.

CIRCLE NO. 225 ON READER SERVICE CARD

ports. Eight settings are offered from 110 baud to 19.2K baud. Menu-driven utilities from any PC or terminal can configure each port as DCE or DTE.

Fifth Generation Systems. The Logical Connection from Fifth Generation can share printers, plotters, and modems. It is an any-to-any switch with eight ports: four serial, two parallel in, and two parallel out. It also provides automatic parallel-to-serial and serial-to-parallel conversion on all channels and buffering of 256KB or 512KB.

All ports are configured independently using software. Because no switches or jumpers are used, the user never needs to open the box. File transfer is possible using the local mode of a communications program. According to company information, units can be daisy-chained to connect up to 315 ports. The switch also permits multiple simultaneous inputs.

The box has a reset button on the side and two panel indicators: run, which lights during normal operation and flashes when the buffer is full; and status, which flashes when the Logical Connection receives a switching command. Although the Logical Connection is one of the more sophisticated switches, it is surprisingly small; the box measures about 14½-by-5 inches and is only ¾-inch thick.

The Logical Connection is somewhat more complicated to configure than the N-to-N switches, but offers more flexibility. The user makes all configurations through a program run-

VENDORS AND PRODUCT SELECTION

Peripheral-sharing switches and their accessories are available from many sources, including retailers. This sampling of vendors and products is by no means a comprehensive list.

Black Box Corporation
P.O. Box 12800
Pittsburgh, PA 15241
412/746-5530
CIRCLE 331 ON READER SERVICE CARD
Product offerings: Two-way and four-way mechanical switches, Multi-Port Spooler II electronic switch, RS-232 patch boxes, cross-matrix switch

Bravo Communications, Inc.
4460 Partridge Court
San Jose, CA 95121
408/270-4500
CIRCLE 332 ON READER SERVICE CARD
Product offerings: Clustered Print Sharer parallel electronic unbuffered switch and parallel line extender

Digital Products, Inc.
108 Water Street
Watertown, MA 02172
617/924-1680, 800/243-2333
CIRCLE 333 ON READER SERVICE CARD
Product offerings: NetCommander electronic serial/parallel switch

Fifth Generation Systems
11200 Industriplex Blvd.
Baton Rouge, LA 70809
504/291-7221, 800/225-2775
CIRCLE 334 ON READER SERVICE CARD
Product offerings: Logical Connection any-to-any serial/parallel switch

Gandalf Data, Inc.
1020 S. Noel Avenue
Wheeling, IL 60090
312/541-6060, 800/426-6336
CIRCLE 335 ON READER SERVICE CARD
Product offerings: Dovtrex data-over-voice system

Giltronix
3780 Fabian Way
Palo Alto, CA 94303
415/493-1300
CIRCLE 336 ON READER SERVICE CARD
Product offerings: EZQueue electronic serial-to-parallel switches

Global Computer Supplies
45 S. Service Road
Plainview, NY 11803
516/752-2299, 800/845-6225
CIRCLE 337 ON READER SERVICE CARD
Product offerings: Serial and parallel mechanical switches

Inmac
Mail Stop 5060
2465 Augustine Drive
Santa Clara, CA 95054
408/727-1970
(Inmac has 19 different locations across the United States)
CIRCLE 338 ON READER SERVICE CARD
Product offerings: Clear Signal Plus, Parallel Smart Switch, Any-to-any Smart Switch, Quick Patch, Universal Data Buffer

Integrated Marketing Corporation (IMC)
1031 E. Duane
Suite H
Sunnyvale, CA 94086
408/730-1112
800/621-0854 extension 309
CIRCLE 339 ON READER SERVICE CARD
Product offerings: Auto T-Switch and the Data Manager

Intellicom
9259 Eton Avenue
Chatsworth, CA 91311
818/882-8866, 800/992-2882
CIRCLE 340 ON READER SERVICE CARD
Product offerings: Long-Link serial/parallel line converter and serial/parallel line extender

Intra Computer, Inc.
101 W. 31st Street
New York, NY 10001
212/947-5533
CIRCLE 341 ON READER SERVICE CARD
Product offerings: Delta Three-Way Switch, crossover switch

IQ Technologies
11811 N.E. First Street
Bellevue, WA 98005
206/451-0232, 800/227-2817
CIRCLE 342 ON READER SERVICE CARD
Product offerings: Universal RS-232 Smart Cable

Reliable Communications, Inc.
136 S. Wolfe Road
Sunnyvale, CA 94086
408/733-1787, 800/222-0042
CIRCLE 343 ON READER SERVICE CARD
Product offerings: The Sierra Exchange serial electronic switches

Support Systems International
150 S. Second Street
Richmond, CA 94804
415/234-9090
CIRCLE 344 ON READER SERVICE CARD
Product offerings: Multiswitch II parallel electronic switches

Server Technology, Inc.
140 Kifer Court, Suite A
Sunnyvale, CA 94086
408/738-8377, 800/835-1515
CIRCLE 345 ON READER SERVICE CARD
Product offerings: EasyPRINT serial/parallel electronic switches

Western Telematic, Inc. (WTI)
5 Sterling
Irvine, CA 92718
714/586-9950, 800/854-7226
CIRCLE 346 ON READER SERVICE CARD
Product offerings: Lasernet and Smart Switch serial/parallel electronic switches, parallel-to-serial converter

BATRAM™

Your LANSAVER.

See us at COMDEX/Spring '87

The results are in. BATRAM outperforms disk drives!

This ultra-high performance IBM PC/XT/AT compatible mass storage device combines the speed of RAM, the safety of long term battery backup and the security of sophisticated ECC (Error Correction Coding) to give you extremely fast, safe, and reliable storage. And now with exclusive FTS/Caching software, additional speed and protection are added to your local area network.

BATRAM IS 700% FASTER THAN A DISK DRIVE

Your local area network applications, like spreadsheets, data base management, and CAD/CAM, which are all disk intensive, benefit from BATRAM's phenomenal speed.

EXCLUSIVE FAULT TOLERANT/CACHING SOFTWARE

FTS/Caching software speeds up your read/write requests by eliminating I/O bottlenecks. BATRAM does this by caching your most requested hard disk information, therefore allowing the server to access RAM memory rather than the disk drive. BATRAM can service these common requests faster than the disk drive, increasing network throughput.

DATA INTEGRITY INSURED

Since BATRAM also holds all network data writes, BATRAM adds Fault Tolerance to your network storage device. With BATRAM, and your latest network backup, data integrity is ensured even upon total failure of your network hard disk.

BATTERIES PROTECT RAM MEMORY

Power interruptions don't phase BATRAM. Your important data is protected by BATRAM's innovative two-week battery backup. These batteries are continually charged, delivering full power so your data is always safe. You can even unplug BATRAM without losing data for up to two weeks. BATRAM's ECC is further assurance of your datas integrity.

BATRAM IS TOUGH

BATRAM's rugged shock-mounted, no-moving-parts construction is built for use and abuse in tough environments. It is totally unaffected by the shake, rattle, and roll that render disks useless.

NO WORRY MEMORY

Call us now for the whole story on the industry's fastest, safest, and most reliable storage device: BATRAM, the no-worry memory. **Dealer inquiries welcome. Call (408) 729-6700.**

Santa Clara Systems, Inc.
A Novell Company
1610 Berryessa Rd., San Jose, CA 95133
(408) 729-6700 / FAX (408) 259-0778 / TLX 176309 SCS SNJ

IBM PC/XT/AT are trademarks of International Business Machines Inc.

CIRCLE 197 ON READER SERVICE CARD

ning in the PC. To install the Logical Connection, the user first connects the PC serial port to the first serial port on the Logical Connection (port number 0) using a cable provided by the manufacturer. This connection is used to download the configuration information to the Logical Connection.

Second, the user defines the physical connections, assigning a name to each device, such as HP Laser or IBM AT, and then sets parameters for the serial ports, such as baud rate, handshaking protocol, and time-out. Parallel ports are simpler—the user is required to set no parameters, but merely sets the automatic form feed to yes or no and assigns a time-out.

Third, the user defines logical connections—how ports will be paired—and downloads this configuration information to the switch. Because the configuration is stored in battery-backed RAM in the Logical Connection, the switch can be left unplugged for an extended period without having to reconfigure. Finally, the user physically connects the devices.

Integrated Marketing Corporation. IMC offers two lines of N-to-N switches: the Auto-T-Switch and the Data Manager Plus 256. The Auto-T-Switch is available in a number of N-to-N configurations: three-to-one, all serial; six-to-one, all serial; three-to-one, all parallel; six-to-one, all parallel; six-to-one with three serial in, three parallel in, and one serial out; and six-to-one with three serial in, three parallel in, and one parallel out. The three-to-one parallel switch was evaluated for this article.

The Auto-T-Switch provides no internal buffering and does not support multiple simultaneous inputs. On models with both serial and parallel ports, the switch performs serial-to-parallel and parallel-to-serial conversion.

Time-out is selectable at 10, 20, 30, 60, 120, or 300 seconds. Bidirectional data flow allows support for modems, plotters, and other PCs. The input device must initiate port selection. For serial ports, the switch has an adjustable baud rate, data bits, stop bits, and parity. It is also possible to configure each port individually.

All configuration is accomplished using jumpers inside the box; no software is necessary. The Auto-T-Switch has no panel controls. Panel indicators include a power light, an on-line light, which indicates that the output device is available to the computer, and one input light for each port.

Serial ports are DB-25 male for output and DB-25 female for input. Because all parallel ports are female Centronics, the switch end of the cable must be male Centronics for parallel input to the switch—this is the typical cable used for connecting a PC to a printer. For parallel output from the switch to the printer, the switch end is male Centronics. The printer used for testing also required a male Centronics connector (as do most IBM parallel printers). The required cable had to be custom-built because it was not available commercially.

The Data Manager Plus 256, also from IMC, is a five-to-one switch with 256KB of standard buffering, expandable to 1MB. It supports multiple simultaneous inputs, and all ports are serial and unidirectional, interfaced through female DB25 connectors. The Data Manager Plus supports data rates up to 38.4K bits per second. Baud rate, data bits, stop bits, and parity are all adjustable by individual port through-switches inside the box.

Using DIP switches, the Data Manager Plus can be hardware-configured optionally to send a form feed at the

The Most Powerful LAN Fits on a Disk.

Network Power. You knew that someday there would be a powerful LAN that didn't need old-technology network boards. It would be fast, easy to install, and run 99% of PC-DOS software. It would be expandable, provide remote access, password-protection, and enable you to use inexpensive terminals as workstations in a PC-DOS environment.

Dream no more, because the power is here.

Its name is LANLink™.

A Software-Driven LAN Powerful Enough To Use RS-232 Ports for Network Communications. In development for over three years, LANLink™ represents the next generation of local area networks. All of the logic which has traditionally resided on network boards is on LANLink's Satellite and Server Diskettes.

No additional hardware is required. Inexpensive serial ports replace "Kilobuck" Network Interface Boards making installation costs one-third that of a board-driven network.

How To Configure a Smart Network...With Dumb Terminals, But Without Dedicated Servers. Boasting a wide variety of configurations, LANLink™ is most often set up as a "Star" having up to eight satellites connected to a central, nondedicated server. Larger networks can have multiple servers, supporting a total of 73 or more network users.

R-LAN™ (Remote-LAN) gives users the ability to interact with a LANLink™ network in real time via modem. Plus, if MultiLink Advanced™ is run on a Satellite, inexpensive dumb terminals can be used to access network disks, files, and programs.

99% of PC-DOS Applications Run In a Totally-Transparent Network Environment. If you know DOS, you already know how to use LANLink™. COPY transfers files among users, and a 2-drive PC Satellite boots 1-2-3 from the Server's hard disk with the entry c:lotus. Each satellite's access can be limited to specific disks, printers, and subdirectories. A wide variety of software including Lotus 1-2-3, dBASE III, and WordStar 2000 is fully compatible. LANLink™ has a collision-free data transfer rate which exceeds 115,000 BPS.

Power Up Your PCs Today. For complete details and the authorized dealer nearest you, call The Software Link TODAY. The LANLink™ Starter Kit is $495 and includes modules for both a Server and a Satellite. For a limited time, 50 feet of shielded RS-232 cable will be included free of charge. Additional Satellite Modules are only $99, each.

LANLink™ is immediately available and comes with a money-back guarantee. VISA, MC, AMEX accepted.

LANLink™

THE SOFTWARE LINK
Developers of LANLink™ and MultiLink® Advanced

3577 Parkway Lane, Atlanta, Georgia 30092
Telex 4996147 SWLINK FAX 404/263-6474

(For the dealer nearest you) **CALL: 800/451-LINK** In Georgia: 404/448-LINK

OEM/Int'l Sales: 404/263-1006 Resellers/VARs: 404/448-5465
Dealer Inquiries Invited

THE SOFTWARE LINK/CANADA 250 Cochrane Drive, Suite 12
Unionville, Ont. L3R 8E5 FAX: 416/477-3234 CALL: 800/387-0453

CIRCLE NO. 196 ON READER SERVICE CARD

end of each job. It also can send the necessary characters to reset an HP LaserJet. Time-out is selectable at 15, 30, 60, or 120 seconds. Units can be daisy-chained to connect as many as 25 inputs to one output.

Reliable Communications, Inc. The Sierra Exchange from Reliable Communications, Inc. is available in two N-to-N configurations: four-to-two (which was tested here), and five-to-one. All ports are RS-232 with female DB-25 connectors. Standard internal buffering is 64KB, expandable to 2MB. The switch allows multiple simultaneous inputs.

Like the IMC Data Manager Plus, the Sierra Exchange supports jumper-selectable data rates up to 38.4K bits per second, adjustable data bits, stop bits, and parity. Each port is configured individually using switches inside the box, and ports are unidirectional rather than bidirectional.

A DIP switch inside the box sets a default output device for each input port; sending control codes to the switch changes this default. The Sierra optionally sends a form feed at the end of each print job; this feature is also selectable through DIP switches.

The Sierra switch has a fixed time-out of five to eight seconds, just one panel control, and a reset button. Six status lights show which ports are active. A flashing light indicates a buffer overflow or *framing error*—errors in stop or start bits. Two lights indicate how much buffer memory is in use.

Support Systems International. Support Systems offers the MultiSwitch II, Model AS-402, which is a four-to-two N-to-N switch, all parallel; all ports are female DB25. Internal buffering is 64, 128, 256, or 512KB. Multiple simultaneous inputs are not supported.

The MultiSwitch offers extensive panel controls, including reset, which clears memory, tests the CPU, and resets options to defaults; pause printing, printer A/B, which changes the output device; page 1, which reprints from the beginning; and copies, for multiple copies. Some panel controls apply only when ASCII characters are sent through the switch, because they look for the form-feed character (FF, ASCII value 12). These include pause top of form, which pauses at the top of the next page; up page, which starts printing one page after the pause point; down page, which starts printing one page before the pause point; and change the number of lines per page.

Panel indicators include power; ready, indicating the MultiSwitch is ready; printer A selected; printer B selected; pause printing; pause top of form; pages in memory; printing page, which displays the number of characters, pages, or copies to be printed; four channel lights, to indicate which input port is active; and busy, which indicates the MultiSwitch is receiving data. Both the printing-page and the pages-in-memory indicators are numeric displays. The printing-page indicator also shows the out-of-paper error message and briefly displays a status message when the automatic form-feed function is turned on or off.

Printer codes control four functions: select printer A, select printer B, pause printing, and reset memory. The MultiSwitch also offers a memory test.

Western Telematic, Inc. WTI's Lasernet is an eight-to-two switch with eight serial inputs, one serial output, and one parallel output; a four-to-two switch is also available. Input connectors are all female DB25; output connectors are one female DB25 and one female Centronics. Lasernet has a 256KB internal buffer and supports multiple simultaneous inputs. WTI also offers two unbuffered models, PSU-41 and PSU-81. Serial-to-parallel conversion is performed internally.

Unlike the other switches described in this article, all the input

ports for the Lasernet must have identical configurations for baud rate, parity, and flow control; the output serial port is configured separately. Again unlike all of the others, the Lasernet's configuration switches are conveniently located on the outside of the box available for a user's easy access.

The time-out is selectable at 250 milliseconds, or 1, 5, or 15 seconds. Panel controls include *clear*, to clear switch memory; *pause*; and TOF (top of form). Panel indicators include eight input indicators that show which port is active; *wait*, which indicates pause status; *mem*, which is off when the buffer is empty, blinks once when the buffer is 10 percent full, blinks twice when the buffer is 20 percent full, and so on; and *rdy*, which indicates the printer is connected and on-line. The Lasernet also offers a self test.

The newer models, PSU-82C and PSU-42C, offer individual configuration of input ports, up to 2MB of internal buffering, automatic TOF, and an end-of-job string that resets the printer to the default parameters just as soon as a job has been completed.

UNCLASSIFIED EQUIPMENT

Several types of peripheral-sharing equipment do not fit neatly into either the N-to-N or any-to-any classification. Inmac offers the Quick Patch, a box with two rows of 16 RJ-11 (standard modular telephone) jacks on the front and 16 RS-232 ports on the back. The RS-232 devices remain plugged into the back of the box, and the user connects and disconnects them by moving RJ-11 cords on the front of the box.

Intra Computer sells a Delta three-way switch that connects any two of three serial devices. For example, the user might connect a computer to a printer in one position, a computer to a modem in a second position, and the modem to the printer for printing from a remote location in a third position. The switch also has a position for no connection at all.

Local telephone companies often lease data-over-voice-encoder (DOVE) systems, with the central unit and a data switch in the telephone company's central office. This *central-office LAN* system duplicates the functionality of an any-to-any switch.

Also called a *data-voice modem* (DVM), a DOVE is a multiplexer that allows voice and data signals to share a standard twisted-pair telephone wire. This wire is already in place for the telephone system in most companies. It is relatively easy to install compared with the coaxial cable or higher-grade data cabling often used for LANs.

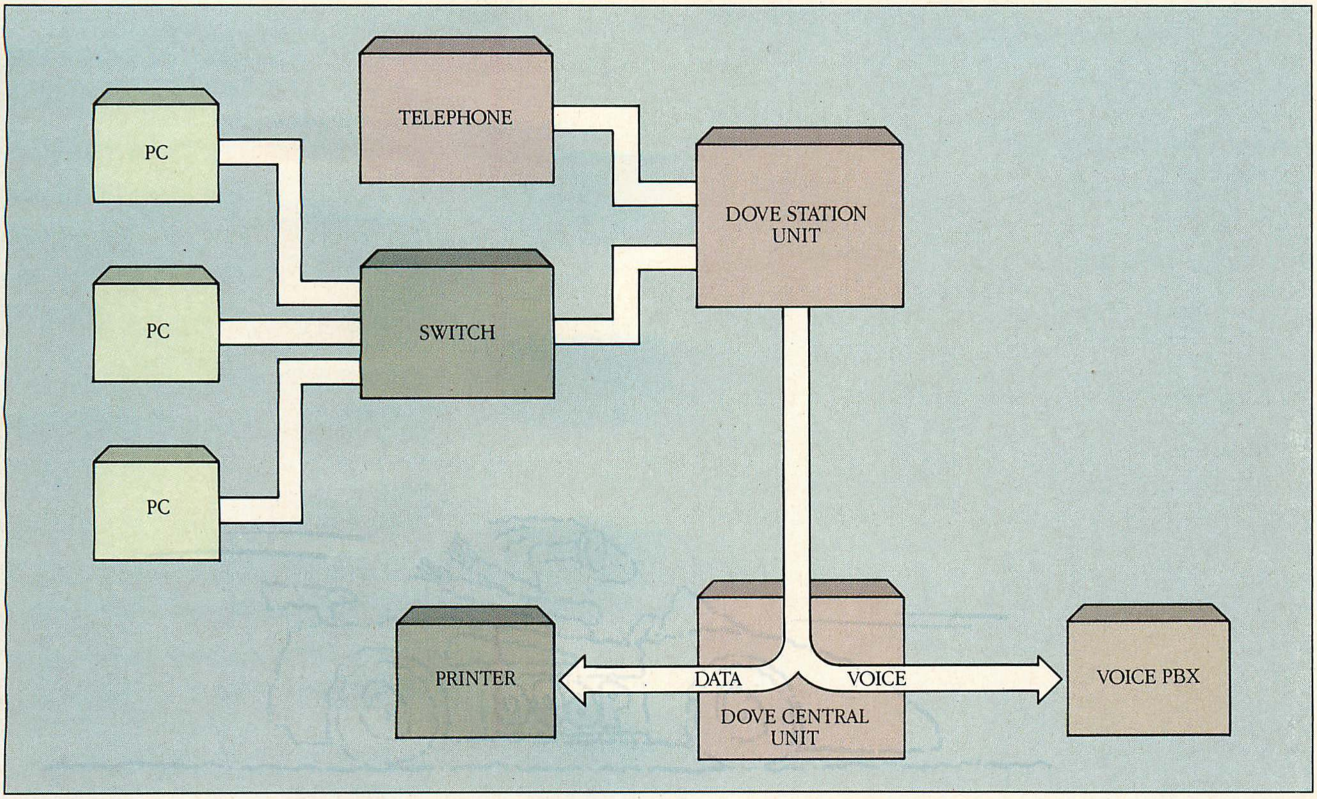
A DOVE modulates data onto two high-frequency carriers, one for send and one for receive. Voice-band signals (speech, dial tone, ring signals, and busy signals) are not affected. PBX and telephone central-office functions are also unaffected, because data signals are stripped off and sent to their destinations without passing through the PBX. Only the wire is shared (see figure 4). Central units are usually rack-mounted, with channel cards fitting into a chassis. Each channel card can accommodate two to four channels.

However, a DOVE lacks the speed of most LANs—DOVE speeds are typically limited to 19.2K baud. Prices vary depending on the region, quantity ordered, length of time contracted to keep the system, and physical distance from the central office.

EXTENDING CONVENIENCE

Line extenders and interface converters can make switch-based peripheral sharing more convenient. A line extender is a small box that conditions and/or amplifies signals or converts them to another form so that they carry farther; line extenders are available for both parallel and serial ports.

FIGURE 4: *Data-over-voice Encoder*



A data-over-voice encoder (DOVE) allows data and voice to share the same wire, simplifying data connections. Only the wiring is shared. DOVE central units often include data-switching capabilities, eliminating the need for an external data switch.

Bravo Communications, Inc. offers a parallel line extender that can support distances up to 3,000 feet under certain circumstances. Although it is designed specifically to work with Bravo's Clustered Print Sharer (CPS), a nonbuffered printer-sharing device, the extender works with any parallel interface. When the extender is used with the CPS, internal signal conditioning allows switch-to-computer distances up to 100 feet and switch-to-printer distances up to 50 feet. The line extender also shares the CPS power supply.

Intellicom's Long-Link converts signals from parallel to serial and back again. It conditions and amplifies the signal for a potential 7,000-foot reach.

Parallel-to-serial and serial-to-parallel converters are useful when a switch does not have the right kind of input or output port for the application. Microlator's Adapta PII-210 converts parallel signals to serial, and serial signals to parallel. Converters also can be combined with buffers. Inmac's Universal Data Buffer combines parallel-to-serial and serial-to-parallel conversion with either a 64KB or 256KB buffer.

Because manufacturers do not always follow the RS-232 specification precisely on serial interfaces, the user may encounter problems when trying to wire connector pins. IQ Technologies' Universal Smart Cable makes it easy to try various combinations of pin connections—it was used for the test set-ups for this article.

The Smart Cable is reconfigured by changing a set of three switches. Unfortunately, IQ Technologies does not document the actual wiring connections made by the switches, so determining which configuration is needed once the connection is working is not easy. The manual does provide a sequence for trying the various switch combinations and lists the most common configurations first.

The user cannot try all possible combinations, however. Western Telematic's Lasernet, for example, requires an input cable configuration not supported by the Smart Cable.

Any configuration can be obtained with a patch box. When interfacing unusual RS-232 devices, a patch box can be a godsend—once the configuration has been determined, the patch-box wiring can be used as a guide to create a custom cable.

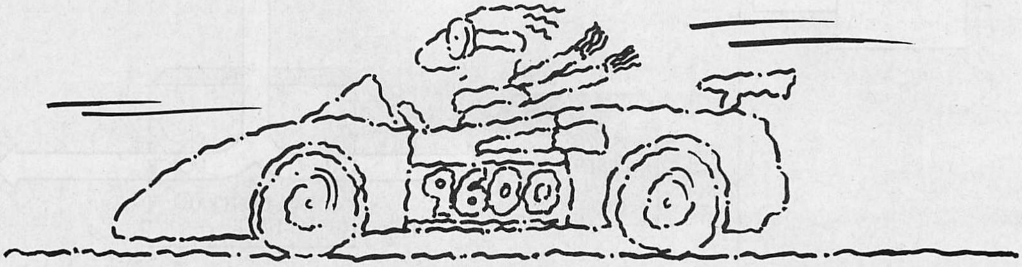
A typical patch box consists of two DB-25 connectors, one male and one female. A small jack is provided for each line on both connectors. Short patch wires are inserted between the jacks for any two pins that are to be connected. LEDs can be wired into the circuit to test the state of a line.

SUITABLE SWITCH

The vast selection of switches—mechanical or electronic, number of connectors, type of cable, internal buffer or not, multiple simultaneous inputs or not—may be overwhelming to the user who is interested merely in running one printer from more than one computer. Nonetheless, this user is much better off wading through all these options than blithely accepting a LAN. For the price and relative ease of installation, switches are much more suitable than LANs.

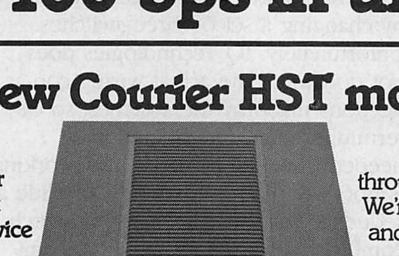
As a peripheral-sharing network grows, a true LAN becomes more attractive. However, both mechanical and electronic switches, enhanced by extenders and other means, can be viable alternatives for users who have small peripheral-sharing needs.

Michael Hurwicz is a consultant and freelance writer who specializes in LANs. He is the author of the book Inside APPC, *available from Architecture Technology.*



Now go 17,400 bps in under $1000.

The new Courier HST modem.



PC Magazine calls USRobotics' Courier HST,™ "the top price-performance ratio at 9,600 bps." Now we've made it almost twice as fast without raising the price.

Our new full-duplex High Speed Technology dial-up modem with MNP™ Level 5 data compression is totally AT command set compatible, delivering data at up to 17,400 bps. And automatic fall-back/fall-forward always assures the fastest speed possible. Courier HST reduces phone line costs every time you use it.

When you buy an HST modem, you get USRobotics' more than 10 years of modem experience gained through sales of over half-a-million modems. We're an organization that will support you and grow with your needs.

PC Week says, "USRobotics' Courier HST produced the most impressive test results based on the modem's suitability to both file transfer and interactive applications."

Other modems talk about speed. Courier HST delivers. Get your hands on one... *fast!*

Free Warranty Extension! Buy an HST before December 31, 1987 and receive a 3-year warranty extension FREE. A $50 value.*

Call 1-800-DIAL-USR Ext. 35

In Illinois (312) 982-5001

USRobotics™

The Intelligent Choice in Data Communications.
8100 McCormick Blvd., Skokie, Illinois 60076.

MNP is the Microcom, Inc. trademark for Microcom Networking Protocol, a public domain error-control protocol.

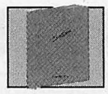
*Total warranty including extension: 5 years.

CIRCLE NO. 213 ON READER SERVICE CARD

Reviews and Updates

SIMPLE
Accuphase Ltd.

TASKVIEW
Sunny Hill Software

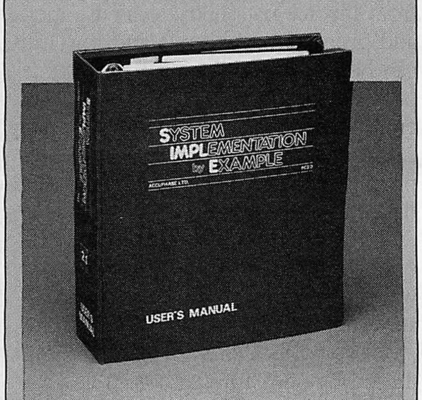


TOPDOS 2.0
FrontRunner Development

SIMPLE

Accuphase Ltd.
Distributed by Software Merchants Unlimited, Inc.
2252 Fillmore Street, Suite 401
San Francisco, CA 94115
800/874-6753; in California, 800/826-3069

PRICE: $395.00, SIMPLE
$49.95, SIMPLER (runtime)



CIRCLE 357 ON READER SERVICE CARD

SIMPLE (System IMPLementation by Example), a data management program from Accuphase Ltd, claims to be the "first picture-oriented application generator." With an eye toward increasing productivity, it eliminates the need to write procedures or to code application programs. The program developer provides the specifications by drawing images on the screen and interacting with worksheets. SIMPLE takes it from there and generates the actual application program. The underlying concept of SIMPLE is unique and potentially useful. However, the product in its current release has several problems and requires such a substantial learning effort that SIMPLE belies its name.

Traditionally, data management application programs are developed using procedural languages, such as COBOL, C, and dBASE. With a procedural language, the developer writes a program that lists the exact steps required to accomplish a task.

However, with a nonprocedural language, such as SQL (structured query language), the developer creates programs of language statements that define the characteristics of the desired resultant set of data. Then the data manager determines the process required to produce that data.

SIMPLE is one such nonprocedural language. The developer communicates to SIMPLE using a query-by-example (QBE) technique to specify the desired programs via a series of screens. According to the SIMPLE documentation, "The principle behind SIMPLE is simple: paint examples of what you want done, then let the computer figure out how to accomplish the task." This program specification approach is designed to be easier to learn and to use than procedural programs or command files.

The basic SIMPLE development screen, called a worksheet, is similar to a spreadsheet screen. Option menus appear across the top of the screen. The user selects options either by typing the first letter of the desired option or by highlighting the option using a reverse-video bar.

The seven options, called modes, of the main menu are DEVELOP, LOAD, MAKEMENU, USEMENU, OTHERS, TUTORIAL, and QUIT. The OTHERS mode offers a utility submenu to set the date, to list a program, to set colors, or to select or specify up to four SIMPLE paths. A SIMPLE path is a mapping of a DOS directory path to one of four SIMPLE logical paths (1 to 4). SIMPLE uses these paths to search for programs and data as it performs operations.

All development is accomplished in the DEVELOP mode. This mode offers a large tree of four additional levels of option menus. The first level of menu options are ADD, REPORT, EDIT, PROCESS, DELETE, IMPORT, and EXPORT. Both IMPORT and EXPORT have submenus for the selection of files and file types for import/export operations. The other five options all lead to a common submenu of options: PERFORM, DESIGN, SPECIFY, FILE, OUTPUT (REPORT only), and CLEAR.

Although more than one menu path can lead to the same apparent action, the path becomes part of the mode. With SPECIFY, for example, the developer can select DEVELOP/ADD/SPECIFY to enter specifications to add data to a file and DEVELOP/EDIT/SPECIFY to enter specifications for a data modification program. The SIMPLE screen is the same in both cases (except for the display line at the top), so the developer must remain aware of the mode path to keep track of the current development effort.

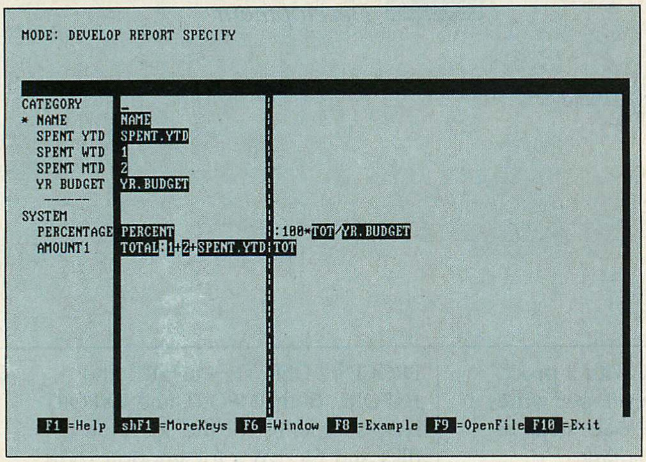
One confusing feature of the DEVELOP mode is that file definition options are available on menus only after the DEVELOP/ADD, REPORT, EDIT, PROCESS, or DELETE modes have been selected. As soon as the DEVELOP mode is chosen, the developer wants to be able to select a file.

The DEVELOP mode is SIMPLE's program generator. The lists of options indicate the types of programs that can be generated, and the DESIGN and SPECIFY modes provide the facilities for designing the programs.

The DESIGN option is used to design screens and report layouts. Editing layouts within this mode is straightforward, although painting a report wider than 80 columns is awkward. Instead of horizontally scrolling a window on a wide report layout screen, the developer must specify multiple screen lines per output line and contend with line wrap boundaries within the 80-column screen.

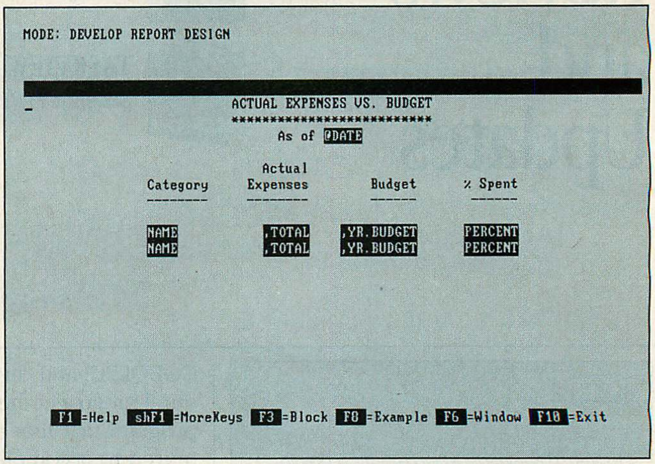
Using the SPECIFY worksheet, the developer writes the processing logic for the program in the form of QBE-

PHOTO 1: *A Specify Worksheet*



The category and system files are opened and the fields to be included in the report are entered in the first column of the worksheet. Data elements, formulas, and conditionals may be entered in successive columns on the screen.

PHOTO 2: *A Report Design Worksheet*



Using the REPORT DESIGN option, the report form is painted on the screen in the desired format and the data elements from the specify worksheet are entered where the actual values will be printed when the report is run.

like examples. This is the heart of the SIMPLE concept. In SPECIFY mode, example elements are declared. These alphanumeric strings are used as variables in the DESIGN and SPECIFY worksheets to hold values, to link files, to link the DESIGN and SPECIFY worksheets, and to refer to data fields.

As expected in the relational approach, two or more files can be opened and linked. In SIMPLE, linkage is indicated by equating example elements of the linking fields in each file.

Formulas and query conditions also are stated on the SPECIFY worksheet, as are marks to indicate whether data can be entered into a field. The SPECIFY worksheet is the mechanism that tells SIMPLE how to manipulate file data, so all DESIGN worksheets for ADD, EDIT, DELETE, and REPORT programs must have associated SPECIFY worksheets. PROCESS, IMPORT, and EXPORT programs do not have DESIGN worksheets, but must have SPECIFY worksheets.

SPECIFY worksheets are laid out in columns and rows. The sheet can be partitioned into a maximum of 8 columns, which must fit within the screen's 80-character width. In the first column, fields are listed vertically for open files (fields do not need to be listed if they will not be used). Example elements, conditional statements, formulas, commands, and updating statements are placed in the second and successive columns (see photo 1).

Example elements are used to refer to data fields. A caret (^) preceding an example element indicates data can be entered into that field. Conditional statements filter data from files, validate data entered or processed, and specify conditions for data update. They consist of values and logical operators (similar to the IF statements in most procedural languages). Depending on the partition column where the conditional statement is placed, data can be selected for processing or optional updating. Commands are used to indicate special handling of records and to create error messages for data-entry programs. The available commands are: \NEW (create new records); \DELETE (delete existing records); \RECALL (undelete records); \NONE (skip record); \ASC and \DES (specify ascending or descending order of indexed files); and \ERR# (create error message).

Data-entry screens incorporate up to four sections: Form, Search, Calls, and Help. The Form section contains the form name, text, prompts, and data entry fields. The developer can arrange the entry prompts in any way by painting the screen. Data can be entered into multiple files. However, a SPECIFY worksheet must list all the corresponding files and all example elements marked with a caret for data entry.

The Search section of a form consists of one or more pop-up windows that allow the user to scroll through data from linked files and to select data for entry into fields. This search window is limited to a single line that SIMPLE positions at the top of the screen.

The Calls section permits the developer to specify optional calls to other programs when the data-entry program is used from a menu. For example, options can be established to allow the data-entry operator to switch from invoice data entry to a customer add program and back. Mandatory calls can be used to execute other SIMPLE programs at desired points in the data-entry screen (for example, a mandatory call can be used to execute an invoice print program at the end of an invoice data-entry screen).

The Help section allows the developer to enter text to be displayed when the end user presses F2 (the application help key).

Using the REPORT DESIGN option, the developer can paint a sample page of a report by typing example elements from any file to represent the data (see photo 2). An associated SPECIFY worksheet must exist. Five statistical functions are available, as well as DATE and PAGE functions. In any DESIGN mode, SIMPLE will provide a default design form with one keystroke.

The MAKEMENU option allows the developer to create a menu for running an application. Each menu choice can be followed by 1 to 19 command lines, which SIMPLE will execute when that option is selected. This proved to be an extremely convenient and easy feature to use.

A substantial amount of power is available in the SIMPLE program specification concept. For example, the SPECIFY worksheet can link together as many as five files, enabling the developer to design a screen that will allow data entry into multiple files simultaneously. Data from multiple files can also

be browsed and included in reports. The joins are done by SIMPLE provided a corresponding SPECIFY worksheet includes all the files needed. In addition, the same SPECIFY worksheet can be used for programs of different types. For example, the developer might use DESIGN and SPECIFY worksheets to create a data-entry program in the DEVELOP/ADD mode, then use the same worksheets to create a data-modification program in the DEVELOP/EDIT mode. Unfortunately, with the current implementation and documentation, it is not easy to learn how to specify operations beyond typical rudimentary file-processing tasks.

Two serious problems and several minor annoyances were encountered when testing SIMPLE, both versions 2.1 and 2.2. First, in accordance with the manual, the installation of the program sets the computer's CONFIG.SYS file specifications to FILES=20 and BUFFERS=1. Although Software Merchants Unlimited's representatives say the program can operate correctly with buffers set to a number greater than 1 (and this does appear to be the case), this fact is not documented in the manual or in the installation procedure. A buffer setting of 1 has a serious impact on the performance of many programs and SIMPLE's requirement that this setting be in the user's CONFIG.SYS file is uniquely and extraordinarily restrictive.

Second, SIMPLE consistently modified the ASCII files from which it was importing data. Each input file grew by 512 bytes; most appeared to be data from the end of the file. It seems that SIMPLE appends the last 512-byte buffer from the input file to the end of the ASCII file. For example, the *PC Tech Journal* test file, ISSUE.ASC, is a 1,311-byte ASCII file of data in delimited format. After importing this file to a SIMPLE file definition, the original ISSUE.ASC file was 1,823 bytes long, with a new DOS date/time stamp. The number of buffers specified in the CONFIG.SYS file (BUFFERS=1 in one test and BUFFERS=20 in another) did not affect this. All three ASCII test files imported exhibited the same 512-byte growth. This type of egregious software error in a production program raises suspicions about the overall quality of the system programming and testing.

One minor annoyance was noted when the *PC Tech Journal* standard data-entry screen was being created. SIMPLE permits the use of graphic characters in screen design; it assigns several of them to keys on the keyboard. All 11 of the single-line box and joint characters are available. However, only 6 of the 11 double-line box characters were available, and only 2 of the 18 single-to-double and double-to-single joints were available. Not only were these and other graphic characters not assigned to keys on the keyboard, but they were not available in any manner. No complete ASCII character table was provided, and characters could not be entered via the standard approach using the Alt key in combination with the numeric keypad digits to enter ASCII character values.

Another annoyance involves the function of the Ctrl-Home key combination, which often is used in other programs to move the cursor to the top left corner of the screen. Pressing Ctrl-Home when in a SPECIFY worksheet erases the entire worksheet without warning (other than the warning given in the reference manual). This is more than a little unfriendly to a new SIMPLE developer.

SIMPLE's tutorial materials do not have adequate depth to support the developer's initial efforts to learn the system. A rudimentary on-line tutorial provides scenarios that present the basics, but they do not address the program's more advanced concepts. The tutorial is better suited to end users learning how to perform basic tasks than to developers trying to learn the capabilities of the program. Additional examples of application system development would be very useful.

A separate tutorial manual (which appeared to be a preliminary version of a future tutorial) also was provided for review. This manual provides additional examples of more sophisticated tasks, such as batch processing and transaction posting within the development of an accounts-receivable system. It is substantially better than the on-line tutorial. However, both the on-line tutorial and the one in the manual merely tell the user what keys to press to complete the lesson; they offer little explanation as to why. It is difficult to apply the tutorial information to other situations and system designs. Some technical explanation as to why a certain approach was taken in each lesson would be valuable, as would page references to the SIMPLE manual.

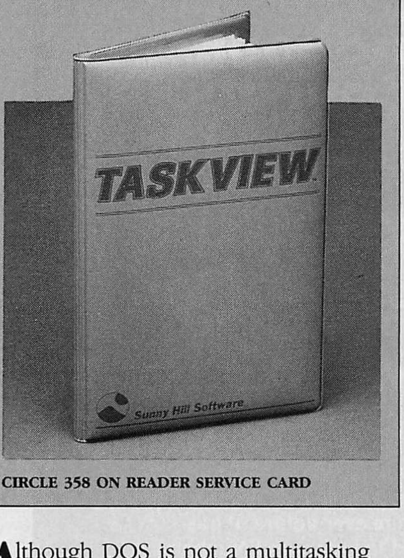
Although not out-and-out bugs, these and other inconsistencies, plus inattention to detail create minor annoyances that make SIMPLE more awkward to use than it should be.

However unique and interesting the SIMPLE concept, the product in its present form has flaws that make it unsuitable for serious application development. The program specification technique is not easy to master for tasks that are more complex and subtle than typical file add, edit, and delete operations. Developers steeped in more traditional methods will find the technique especially foreign and difficult. This is not inherently bad—most tools require a certain amount of learning before their full benefits can be realized. However, a tool that requires a substantial learning effort should provide substantial benefit once learned. Due to its shortcomings, SIMPLE does not provide that benefit. SIMPLE is not consistent in overall design and quality with many of the programs reviewed in *PC Tech Journal's* continuing series on data managers.

—DAVE BROWNING

TASKVIEW
Sunny Hill Software
P.O. Box 33711
Seattle, WA 98133
800/367-0651; in Washington, 206/367-0650

PRICE: $79.95



CIRCLE 358 ON READER SERVICE CARD

Although DOS is not a multitasking operating system, users can bypass this limitation by using an operating environment to mediate between DOS and coresident application programs. The convenience of running several programs at once and switching among them easily has attracted many users to operating environments such as IBM's TopView and Microsoft Windows. Another in this product line is TASKVIEW from Sunny Hill Software.

Unlike task-switching environments that load several programs but run only one at a time, TASKVIEW provides true multitasking for most PC applications. It cycles through the loaded programs, giving each one a fraction of a second to run; it does this so fast that the programs appear to be running simultaneously (although only one appears on the screen at a time).

TASKVIEW is a powerful, yet easy-to-operate environment. Most users will want to take advantage of its menu system, which is convenient and straightforward. When TASKVIEW is invoked, it displays a menu of choices. If "Run a program" is selected, TASKVIEW displays a menu of the programs it has been configured for and runs the one chosen. Operation of the program continues just as it would if it were executed from the DOS command line.

Users can switch among various programs by one of three methods: choosing from a menu, using a hot key, or typing a command at the DOS prompt. At any time, users can invoke the menu by pressing Ctrl-Left Shift-End. From there, users can choose to run another program, to switch to an already-running program, or to reconfigure TASKVIEW. To switch to an already-running program, the user selects "Switch programs." TASKVIEW then displays a list all the active programs, from which the user chooses one.

Because TASKVIEW assigns each running program a number, users can switch quickly among programs by pressing Ctrl-Left Shift and the numeric keypad digit corresponding to the desired program. No matter how it is selected, the desired program appears on the screen and continues running as if TASKVIEW had never intervened.

If the user does not want to use the menu system, TASKVIEW also can be controlled from the DOS command line. The TASKVIEW commands OPEN, SPAWN, and SWITCHTO perform all the functions of the menu-driven system, but take less memory (leaving more memory for applications). However, even expert users may prefer the menu system because it is simple and quick.

TASKVIEW, like TopView and Windows, must be configured for each program that will be run under it. The user fills out a simple on-screen form, providing the program name, filename, starting path, memory requirements, and other basic information. This configuration is necessary only once—when the program is run under TASKVIEW for the first time. The TASKVIEW manual lists all the information needed to configure TASKVIEW for 39 popular programs. Configuring TASKVIEW for other programs should take less than 15 minutes of experimentation, even for the inexperienced user.

TASKVIEW uses virtual memory to allow the computer to run more programs than would fit in RAM at one time. In other words, if the user loads a program when the available memory is insufficient to hold it, TASKVIEW will attempt to make space by swapping out another running program to disk. It will suspend the swapped-out program, giving it no further processor time until the user switches back to it.

Swapping is invisible to the user, except for a short pause that sometimes occurs. With a hard disk, swapping usually takes less than five seconds on a PC; with a RAM disk, swapping is almost instantaneous. Some programs cannot be swapped in TASKVIEW—in particular, communications programs and programs that support a mouse. However, these programs still will run under TASKVIEW.

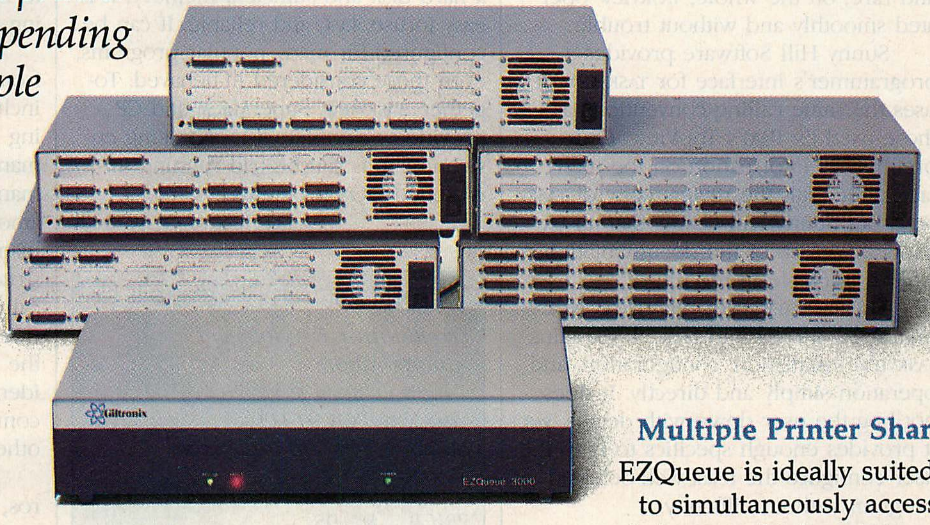
Unlike RoseSoft's ProKey or Borland's SideKick and SuperKey, TASKVIEW does not interfere with other memory-resident programs. It also can run a wide variety of application programs, including many that other operating environments consider ill-behaved. For example, programs that write directly to the screen and read directly from the keyboard are fully supported; in fact, TASKVIEW comes with special loader programs to be used with MicroPro's WordStar 3.3, Nashoba Systems' Nutshell 2.0, Leading Edge Word Processor 1.3a, and BIT Software's BITcom 2.6a. The loader programs allow TASKVIEW to run applications in the background, even though they usually write directly to the screen.

TASKVIEW performs remarkably well, causing almost no noticeable reduction in the performance of each program. Even running four copies of DOS COMMAND.COM (each running a .BAT file) and WordPerfect, all programs performed as if they had total control of the machine. The manual claims that TASKVIEW may actually speed up some DOS operations, but this was not apparent during testing.

Testing revealed only two minor problems with TASKVIEW. First, when running a program configured with insufficient memory, the DOS message "Program too big to fit in memory" appears and disappears before the user has a chance to read it. TASKVIEW itself

EZQueue 3000 . . . . Peripheral Sharing Plus!

The Giltronix EZQueue™ allows up to twenty-four computers to simultaneously share one or more laser printers, dot matrix printers, or plotters. In just a few minutes, you can configure the EZQueue to efficiently manage a single shared printer or multiple printers depending on your need. Ten simple to use and expandable models are offered, each with up to two megabytes of dynamic buffering.



Multiple Printer Sharing

EZQueue is ideally suited to simultaneously access and share multiple printer resources. Each EZQueue is provided with Giltronix GX software, a DOS compatible program that makes selecting a desired printer a snap. Once installed on a computer, a few keystrokes will invoke GX from within any software program to quickly select the desired printer resource. Both serial and parallel interfaces are supported.

Increased Productivity

EZQueue goes beyond the cost advantages of sharing expensive printer resources. In a communication environment, files containing data for printing mailing labels can be directed through EZQueue to a dot matrix printer. At the same time, files containing data for letter quality output requirements can be directed to a specific laser printer containing the desired paper style. Accounting departments can share the various printer resources needed to print invoices or checks. The flexibility of EZQueue can enhance productivity and reduce total system cost in virtually any workgroup environment.

Dynamic Buffering for Simultaneous Inputs

Buffering is dynamically allocated to one or more printer queues allowing simultaneous input from each of the attached computers with no waiting. When a shared printer is busy, files are stacked up in the EZQueue's buffer and output to the designated printer when it is ready to receive the next file.

Simple Installation

No special cables are required because each EZQueue serial port automatically adjusts to signal directions. "Smart" ports, a "21" step installation guide and Giltronix toll-free tech support guarantee a successful installation with minimal effort.

Expandability

Unlike competitive systems, EZQueue's buffer size and number of ports can be expanded as your system grows. A minimum EZQueue starts with six serial ports, two parallel ports and 256K bytes of dynamic buffering. Serial ports can be added in six port increments up to a total of twenty-four serial ports. Buffering can be added in 256K byte increments up to a total of 2 megabytes.

Headquarters
3780 Fabian Way
Palo Alto, CA 94303
(415) 493-1300
Telex 345542

Giltronix

Call Giltronix toll-free at (800) 531-1300
In California, call (800) 521-1330

gives no indication that an error has occurred and behaves as if the program has terminated normally.

The second problem is that TASKVIEW is inconsistent when reporting whether sufficient memory is available to run a program. Sometimes during testing, it would not allow the user to run a program on the first try. Then on a second try immediately afterward, the same program would run without incident. These problems were intermittent and rare; on the whole, TASKVIEW operated smoothly and without trouble.

Sunny Hill Software provides a programmer's interface for TASKVIEW. It uses the same calling conventions as those used by IBM's TopView. Only preliminary information on the interface came with the review copy of TASKVIEW, but Sunny Hill claims that an expanded version of the documentation will be available in the future.

The TASKVIEW user manual is well-organized and easy to read. It explains TASKVIEW installation, configuration, and operation simply and directly. It does not bog the user down with details, yet it provides enough specifics to help the user configure the least-well-behaved program without difficulty.

The minimum requirements for TASKVIEW are two disk drives or one disk drive and a hard disk, DOS 2.0 or later, and 186KB of memory. However, this minimum configuration is hardly practical, and Sunny Hill Software recommends at least 256KB (the more, the better) and a hard disk. TASKVIEW runs on a computer with an EGA, CGA, or monochrome adapter, but it does not support the special features of the Hercules Graphics Card.

TASKVIEW comes with two other programs, SuperMacs and CP. These provide keyboard macros (similar to SuperKey and ProKey) and application-to-application cut and paste (allowing information from the display of one program to be used as input for another). Both SuperMacs and CP are compatible with TASKVIEW.

SuperMacs has a simple pop-up menu user interface, invoked by pressing Alt-Esc. The SuperMacs menu includes the following: choices to Define a new key; Load, Merge, or Save a file of key macros; Clear (erase) all macros; and Write the screen to a file. SuperMacs usage is clearly explained on the screen at all times.

The cut-and-paste capability is available either in SuperMacs or, when only this function is desired, in CP. In both, the user marks the upper-left and lower-right corners of the region to be copied. SuperMacs or CP copies the region into its memory, then recalls it on demand, sending it to the currently running program as if it had been typed in from the keyboard. Both programs insert a carriage return after each line of the region.

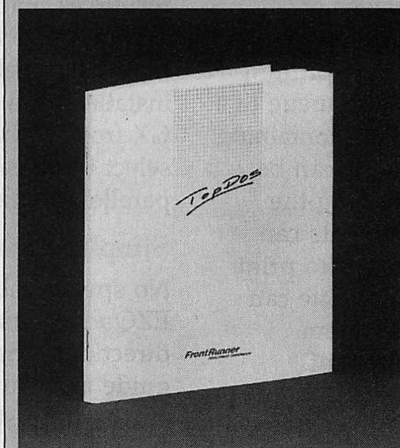
TASKVIEW provides an extremely useful function: it allows users to switch quickly among running programs. It is an excellent product. Given a hard disk and sufficient memory, it is easy to use, fast, and reliable. It can be configured for many popular programs, even those considered ill-behaved. Together, TASKVIEW, SuperMacs, and CP provide as complete a multitasking environment as can be had running unmodified DOS programs.

—ARTHUR A. GLECKLER

TOPDOS 2.0

FrontRunner Development Corporation
14656 Oxnard Street
Van Nuys, CA 91411
800/654-7494, in California, 818/376-1322

PRICE: $69.95



CIRCLE 359 ON READER SERVICE CARD

A major weakness of DOS is its primitive user interface. TopDOS, from FrontRunner Development Corporation, provides an alternative interface to the user by interposing itself between COMMAND.COM and the keyboard, which normally handles user input. TopDOS is not really an entirely new command processor, because it cannot execute batch files or load additional programs; it merely screens keyboard input and passes it to the same standard command processor. In the process, it greatly improves the command-line interface, enhancing existing DOS commands and adding new ones.

TopDOS is a terminate-and-stay-resident (TSR) program that can be loaded from the DOS command line or from a batch file. Once installed, it is disabled and enabled using either a hot key or a DOS command. If it is the most recently loaded TSR, it can be unloaded from memory. And even if TopDOS is loaded, the user can send single characters or commands directly to DOS, bypassing TopDOS, by prefacing them with a bypass character.

Many TopDOS features are derived from the UNIX operating system. These include a history mechanism (for showing and reusing the last several commands entered to DOS); a MOVE command (for moving, not copying, files among directories); a WHEREIS command (for locating files in the directory structure); auto-completion (automatically completing the name of a file or directory on the command line when the user types enough characters to identify it); and aliases (custom DOS commands, used as abbreviations for other commands and their parameters).

TopDOS also has keyboard macros, enhanced command-line editing, on-line help for DOS commands, a RAM-resident text editor, and a TREE program that displays a disk's directory structure and provides point-and-shoot commands to copy, delete, move, edit, and execute files.

TopDOS's command history can save up to 200 characters of commands. The user can display a numbered list of the commands by entering the HISTORY command. Each saved command then can be invoked by number. The command is issued immediately—the user has no opportunity to change it. When invoked in this manner, the command is not added to the history list.

The other method of recalling commands is to scroll through the displayed list using the arrow keys. Each press of the Up arrow key copies the next command in reverse chronological sequence to the command line. The Down arrow key backtracks through the list. Using this method, the user can edit the command before he executes it by pressing the Enter key. A command recalled in this way is entered into the history list.

One of the most useful of TopDOS's features is auto-completion. It is invoked by pressing the Tab key while typing a command or a filename on the DOS command line.

If the user presses Tab while entering a command, TopDOS searches all directories listed in the current DOS path and lists all internal DOS commands and executable files that begin with the letters already entered. For example, pressing d-Tab might yield:

*DATE
*DEL
*DIR
DEBUG.COM
DISKCOMP.COM
DISKCOPY.COM

and any other commands beginning with "d" in the current path. The asterisks indicate internal DOS commands.

If TopDOS finds only one possible completion for the command being entered, it automatically types the characters needed to complete the command. If more than one completion is possible, TopDOS lists them all and allows the user to choose between them, either by typing additional characters or by selecting one of the possible commands using the arrow keys.

Auto-completion also works when a filename is entered as a command-line argument. However, in this case, TopDOS does not search the DOS path for possible completions.

The keyboard macro feature of TopDOS is not in the same league as those in SuperKey and ProKey. TopDOS can define only 11 macros; it assigns them only to the 10 function keys and Alt-=. Command aliases are a second form of macros invoked by name. A TopDOS command displays a list of all aliases and macros.

TopDOS provides full editing features on the command line. Users can move left and right character or by word, insert and delete characters anywhere on the line, and cut and paste within the command line. These features, plus TopDOS's ability to recall rapidly commands entered previously, help the user overcome the often cumbersome command-line interface.

The built-in text editor is an integral part of TopDOS. Because the editor resides in RAM whenever TopDOS is loaded, it starts instantaneously; the user does not have to wait for it to load from disk, unlike most editors.

TopDOS's editor is simple and fast, adequate for its intended use of editing batch files and short text files. It is certainly insufficient, however, for editing programs or long documents. Although it can edit files up to 54KB in size, its command set is somewhat limited. Cut and paste, for example, is confined to one line at a time, and neither search nor replace commands are available.

The TREE command is another comprehensive, built-in feature of TopDOS. This command displays a representation of part of the current disk's directory tree and allows the user to examine various portions by pointing with the arrow keys. The display shows the list of files in the selected directory, plus those in its parent directory, sibling directories, and subdirectories. The user can select files for copying, moving, deleting, executing, or editing.

The TREE command operates smoothly and quickly. It is intuitive and forgiving of mistakes—definitely an improvement over performing the same functions from the DOS command line. However, during testing, the editing feature crashed the computer several times when TREE was invoked from a DOS shell inside an application.

TopDOS provides two utility programs for use as external commands. The first, TREECOPY, copies files from a directory and all its subdirectories, optionally recreating the subdirectory structure at the target. It brings to earlier versions of DOS the same capabilities that XCOPY brings to DOS 3.2 and later. The other utility, CLUSTER, determines the file a given disk cluster belongs to. It is useful for identifying a file with an unreadable cluster.

In addition to these new features, TopDOS enhances the existing built-in DOS commands. For example, commands that operate on a set of files specified by a wild-card pattern can be further limited to operate only on files before or after a given date. Similar to UNIX, any command entered with insufficient or erroneous parameters generates a message describing the correct usage. This message also can be displayed on request.

TopDOS runs on any PC-compatible computer with DOS 2.0 or later. It requires 38KB of RAM when loaded. It is a well-designed, well-executed product. Its command set uses a logical, consistent set of options, and operation is quick and convenient. It comes with an excellent manual and quick-reference card, augmented by on-line help for most operations.

TopDOS is best-suited for expert DOS users, especially those who miss the many conveniences available in larger operating systems. It also can be helpful to anyone who spends a lot of time conversing with DOS by means of its command line.

—ARTHUR A. GLECKLER

The new generation of software development tools are here.

Now . . .

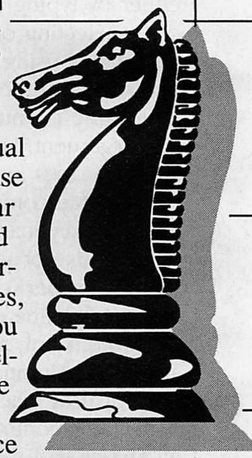
. . . It's Your Move

Arity . . .

The only fully-integrated family of software development tools designed for today's programming needs.

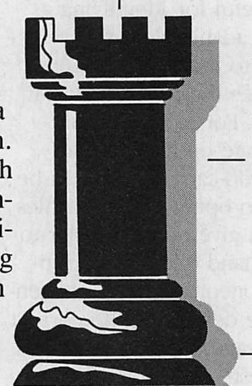
You're looking for a language which can handle today's programming tasks: expert systems, natural language, relational databases, intelligent human interfaces. Your best move is Arity/Prolog. Arity/Prolog is the foundation for a variety of programming tools designed to meet your programming needs.

Prolog



Arity/Prolog is a true superset of the Edinburgh Prolog standard. It includes features such as one gigabyte of virtual memory, complete string support, database partitioning, definite clause grammar support, and full MS-DOS access. And Arity/Prolog has highly-developed interfaces to other programming languages, such as C, assembly, and Pascal. So you don't have to abandon your existing development efforts to take advantage of the power and flexibility of Arity/Prolog. Arity/Prolog is the overwhelming choice of users and reviewers alike as the premier Prolog implementation for IBM PCs and compatibles.

Expert Systems



Arity/Expert is a programming tool which bridges the gap between a human's view of a problem and a computer's view of the problem. Arity/Expert is a frame-based system which features backward chaining, automatic explanation generation, positive and negative confidence factors, and complete system debugging facilities. And Arity/Expert is designed with a unique open-ended architecture that allows you to customize your expert system to match your individual needs.

Check us out!

Call today for more information on the Arity family of software development tools.

1-800-PC-Arity
(Mass: 617-371-1243)

SQL

Structured Query Language (SQL) is fast becoming the industry standard relational database interface language. The Arity/SQL package lets you easily add this familiar database interface to your Arity/Prolog applications. Using Arity/SQL queries, you can easily display specific information from a database table, combine data from many different tables, and perform comparisons among data in the database. If you're planning to incorporate relational database technology into any of your applications, you'll want to use the combination of Arity/Prolog and Arity/SQL to speed your development efforts.

A R I T Y

Arity Corporation 30 Domino Drive
Concord, Massachusetts
01742

CIRCLE NO. 136 ON READER SERVICE CARD

When a Program Has to Work

Formal program verification is a difficult but powerful approach to increasing software reliability.

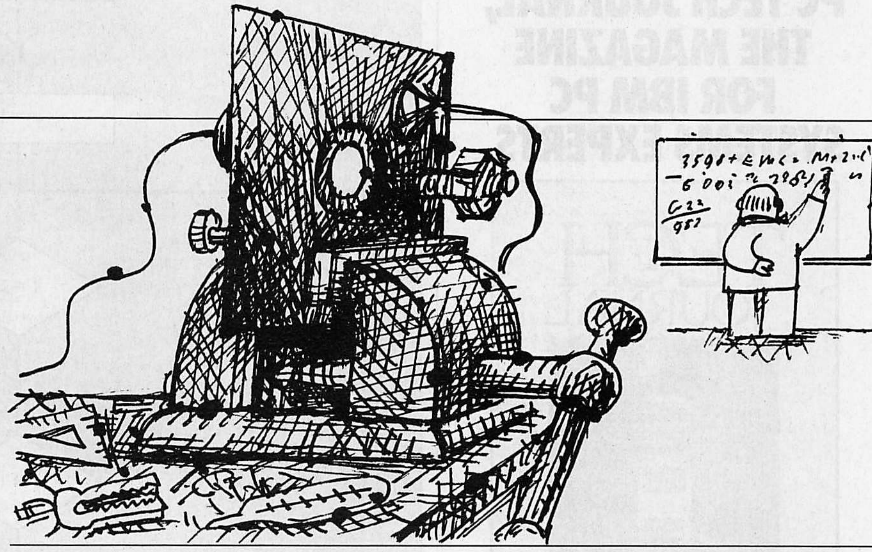


ILLUSTRATION • MACIEK ALBRECHT

Consider the following scenario . . . The ribbon has been cut and the hoopla has subsided; finally it is time to get down to the serious business of bringing the newborn South Bronx Nuclear Power Station up to full power. Some residents and businesses had questioned the wisdom of such a move, but the location really was ideal and the price of the real estate had been right. In any case, the local politicians had seen to it that that the operation of the plant would be controlled completely in the software—thereby ensuring that it would be free of any possibility of human error. Lucky for you, as programmer-in-charge, that you discovered the "last" major bug a good two days before . . .

Or, how about this possibility: Today was the day—the fabrication of the first five million chip sets. Following Big Blue's latest West Coast acquisition, no one was the least bit surprised to learn that the much-heralded MacIn-Dos operating system would require a new chip, the 680386. As the genius behind the microcode, you stare vacantly at the two megatransistor layout pinned to the wall—an amazing feat, considering you had only six months of development time. It *does* work, you tell yourself—after all, it ran Flight Simulator in protected mode, didn't it?

Let's face it, although these scenes are obviously hypothetical (and perhaps, far-fetched), situations are going to come up in which a piece of software just has to work. As any programmer knows, the traditional methods of increasing program reliability—desk checking and testing—are far from perfect. The use of structured programming disciplines and high-level language features can help enormously, of course, but a programmer can always write a structured program that does the wrong thing.

Formal program verification, a difficult but powerful approach to increasing software reliability, is worth considering. Verification, as it is known by researchers, is one of the first and most important applications of machine reasoning. Simply stated, verification is intended to prove in a mathematical sense that a program is correct.

Unlike testing, which can affirm that a program works properly for a finite set of test inputs, formal verification seeks to prove correctness for all possible inputs—even if the possibilities are infinite. Formal verification aspires to a degree of reliability that traditional testing simply cannot match.

Once the most straightforward errors have been exorcised from a program, the remaining bugs are likely to be subtle boundary-condition problems that are difficult to expose using automatic testing. Moreover, for realtime applications, it can be difficult to predict likely inputs in advance. For example, a recent space shuttle mission was aborted just moments before launch because of a software problem. Apparently, the system had never been tested under exact launch conditions.

DEFINING CORRECTNESS

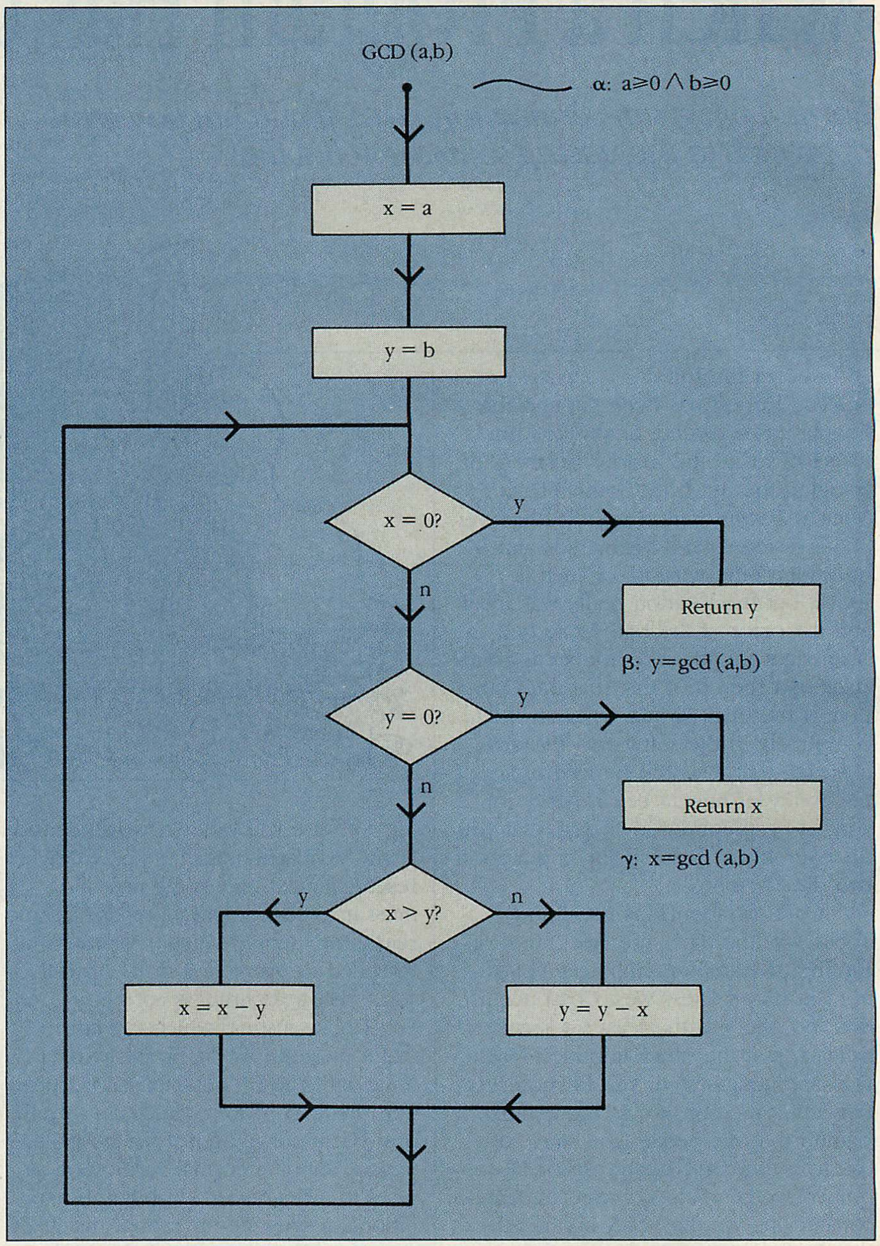
Before describing how formal verification works, we should discuss what it means for a program to be correct. In the first place, correctness has no meaning unless the exact intent of the program in question is understood.

A good example is the DOS 2.x and 3.x call that writes data to a file. If zero is specified as the number of bytes to be written, the file is actually truncated at the current seek position. Although this oddity is not documented in the accompanying DOS manual, many programmers take advantage of it as the only way to return file storage to the operating system short of deleting the file. Should this be considered a bug or a program feature? Only Microsoft can say for sure.

Clearly, to prove the correctness of a program in a mathematical sense, the program's intent must be able to be expressed in a precise and formal manner. This can be accomplished in two ways. One is to exhibit a second program that defines the desired behavior of the first. The second program might be a simpler, less optimal version of the first that can be determined by inspection to do the right thing. Proving the correctness of the first program then becomes a matter of showing that it is equivalent to the second.

The more frequently used method of expressing intent, and the one we will be concerned with here, is to write a description of the program in a formal specification language. The spec-

FIGURE 1: *Flowchart for GCD Algorithm*



In this flowchart, α is the input assertion, β and γ are the output assertions.

ification language can be either a flavor of mathematical logic or a higher-level language that can be translated easily into mathematical logic. In either case, the intent of the program is expressed in a *declarative* manner rather than in a *procedural* manner. In other words, the specification says what the program is supposed to do, whereas the procedural explanation says how it is supposed to do it.

Suppose the program being analyzed were a sorting algorithm that was taking an array *A* of dimension *n* as input. The specification might include an assertion (or logical formula) stating that after execution of the algorithm, the elements of *A* are in ascending order—that is,

$$1 \leq i \wedge i < n \quad \supset \quad A[i] \leq A[i+1]$$

Here, $\wedge$ stands for the logical *and*, and $\supset$ stands for logical *implies*. Note that this assertion says nothing at all about which sorting technique is to be used to accomplish the task—it could be a bubble sort, a quicksort, or whatever. But it is not really enough just to say that the output is sorted. To be complete, another assertion stating that the output is a permutation of the input also would be necessary.

FLOYD'S METHOD

In essence, Floyd's Method of formal verification (which is named after the Stanford University professor who popularized it in the 1960s) works by reducing a program to a set of formulas in mathematical logic that are true if and only if the program is correct. The formulas are proved either manually or with the aid of an automatic theorem-proving system.

Floyd's Method requires the program to be specified by associating assertions at various points within it. In particular, an input assertion is placed at the entry point of the program, and an output assertion is placed at each exit. The input assertion states conditions that the inputs to the program are assumed to satisfy. In the case of a program ROOT that computes square roots, for example, the input assertion might require the input to be a non-negative real number.

The output assertions, on the other hand, say what relationship the outputs of the program must bear to the inputs. If x represents the input to ROOT and y the output, the output assertion might be something like

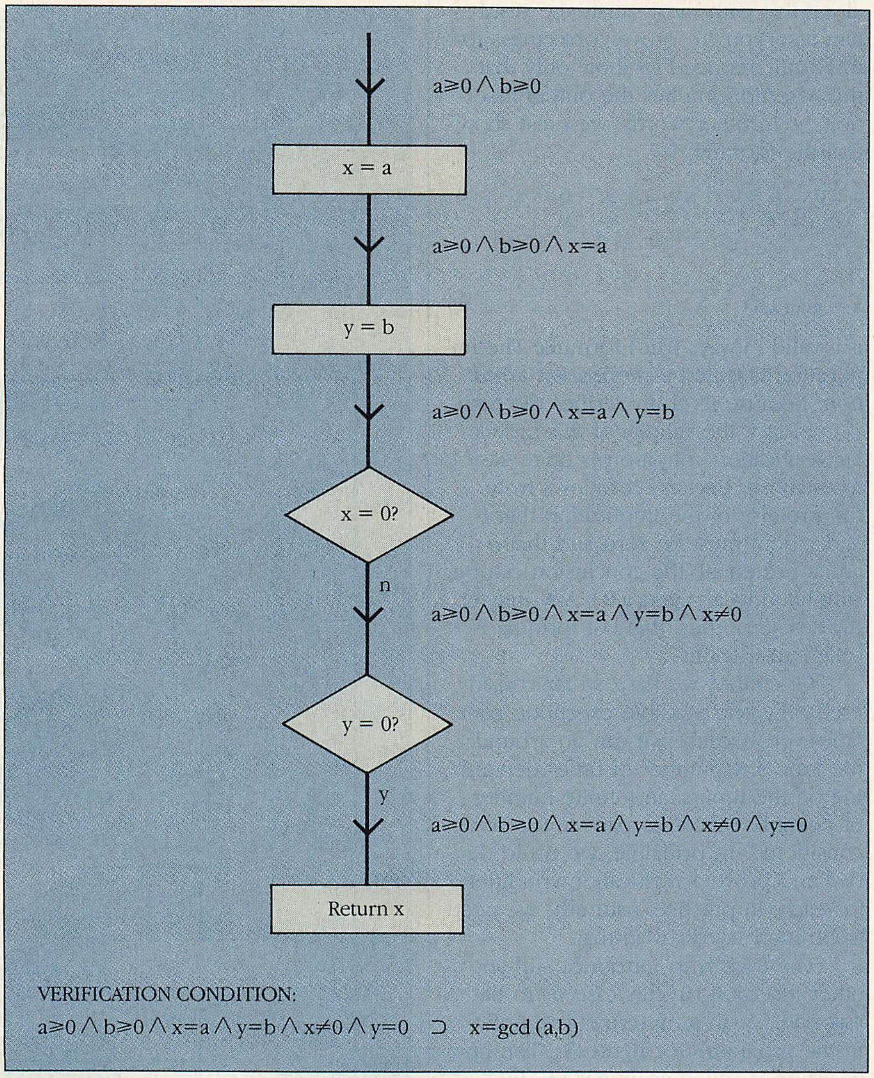
$$| y - \sqrt{x} | < .001 * \sqrt{x}$$

This states that ROOT is good to within .1 percent of the true square root of x. (Floyd's Method also entails that assertions be placed at intermediate points within the program. This process is explained below.)

Now consider the program GCD(a,b) (see the flowchart diagrammed in figure 1), which computes the greatest common divisor (GCD) of two non-negative integers a and b. (For example, the greatest common divisor of 12 and 8 is 4.) Program GCD is a simplified version of the Euclidean Algorithm. As the diagram clearly shows, it works by repeatedly subtracting its inputs from each other until one or the other gets down to zero.

Figure 1 also shows an input assertion annotating the entry point to the flowchart and an output assertion at each of the two exits. The input assertion, $\boldsymbol{\alpha}$, prescribes that the two inputs to the program be non-negative (the algorithm would not always work otherwise), and the output assertions, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, state that the value returned by the program is in fact the greatest common divisor of a and b. (Note that the function **gcd( )** in this specification refers to the true mathematical **gcd** function—not to the program of the same name.)

FIGURE 2: *Symbolic Execution of a Path*



Symbolic execution of a path through the flowchart (in this case y = 0 and x ≠ 0) yields a verification condition that asserts the correctness of the path.

What we want to show is that if particular inputs a and b satisfy the input assertion, then one or the other of the output assertions (depending on which exit is taken) will be true upon return from the program. The demonstration is carried out by means of a technique known as *symbolic execution*. The process of symbolic execution is very much like simulation, except that symbolic inputs are used rather than actual numeric inputs. If program execution can be thought of as pushing numerical inputs around the flowchart, then symbolic execution can be thought of as pushing the input assertion around the flowchart.

Here is how it is done. First consider a straight shot through the flowchart from the entry point through to the second exit, without looping even once. Figure 2 shows this path, along with the results of pushing the input assertion from statement to statement. The head of the path is annotated with the input assertion itself—all we know at the beginning is that both a and b are non-negative.

The effect of symbolically executing the statement **x = a** is to conjoin another clause stating that the variables x and a now have the same value. In similar fashion, pushing this assertion over the assignment **y = b** results in adding a clause stating that y and b now have the same value. Next, passing through the no branch of the test on x, we tack on a clause asserting the result of that test. Finally, moving through the corresponding test on y, we arrive at the resultant assertion that is shown just above the return box.

Any program inputs that satisfy the input assertion and drive execution along this path must satisfy the resultant assertion. To prove correctness for this path, we need to show only that this assertion implies the output assertion, γ. In other words, we must show that the formula

$$a \geq 0 \wedge b \geq 0 \wedge x = a \wedge y = b \wedge x <> 0 \wedge y = 0$$

$$\supset$$

$$x = \gcd(a,b)$$

is a valid (always true) formula. The implication is called a *verification condition*, because its truth verifies the path.

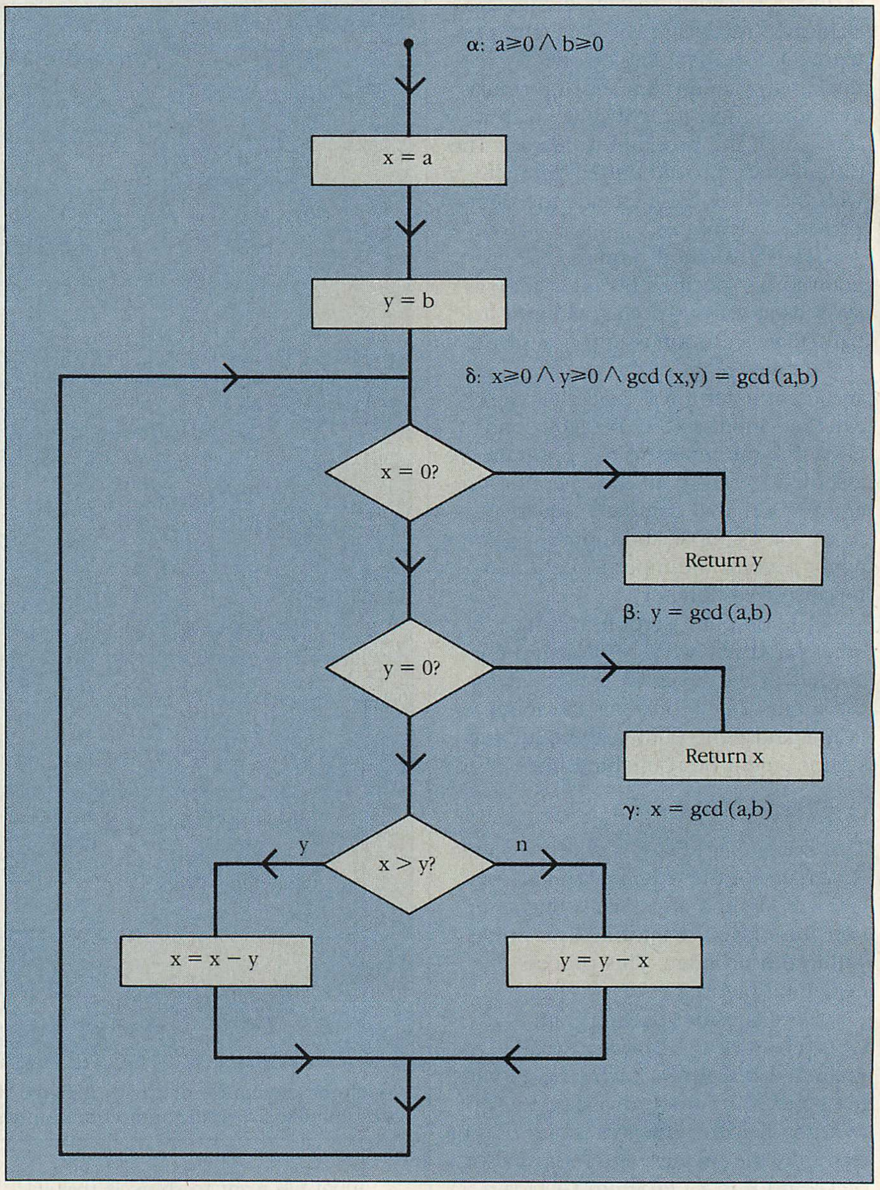
In fact, the validity of this particular verification condition is quite easy to establish. Because it follows from the premise of the implication that *b* and *y* both must be zero and that *a* and *x* are equal, the conclusion can be simplified to $a = \gcd(a,0)$. Any integer divides zero; thus, this last formula is called an identity.

Of course, we have so far considered only one possible execution path. However, because we can go around the loop any number of times depending on the inputs, an infinite number of possible execution paths must be considered. In principle, we could derive and prove a verification condition for each. In practice, naturally, we need to be more clever than that.

The trick is to introduce still another assertion (in this case δ) to be associated with an intermediate point in the program (see figure 3). The new assertion is called an *invariant*, because it is supposed to be true whenever flow of control reaches the point in the program to which it is attached. In the example used here, the invariant states that the variables *x* and *y* are non-negative and that the GCD of the inputs *a* and *b* is the same as the GCD of the variables *x* and *y*.

The invariant is placed strategically at a point where it cuts the main loop of the program. As shown in figure 4, this placement permits the flowchart to be decomposed into a number of strips of straight-line code, each strip beginning and ending with an assertion. Any execution path through the original flowchart can be broken down into a traversal of the strips. (For an execution path that loops, strips II and III might each be traversed more than once.) Note that in the sequence of strips corresponding to a given execution, the assertion labeling the bottom of each strip is the same as the assertion labeling the top of the next strip in the sequence.

FIGURE 3: *Annotated Flowchart*



A new assertion, δ, has been added within the loop in the flowchart. The *loop invariant* δ is true whenever control reaches that point.

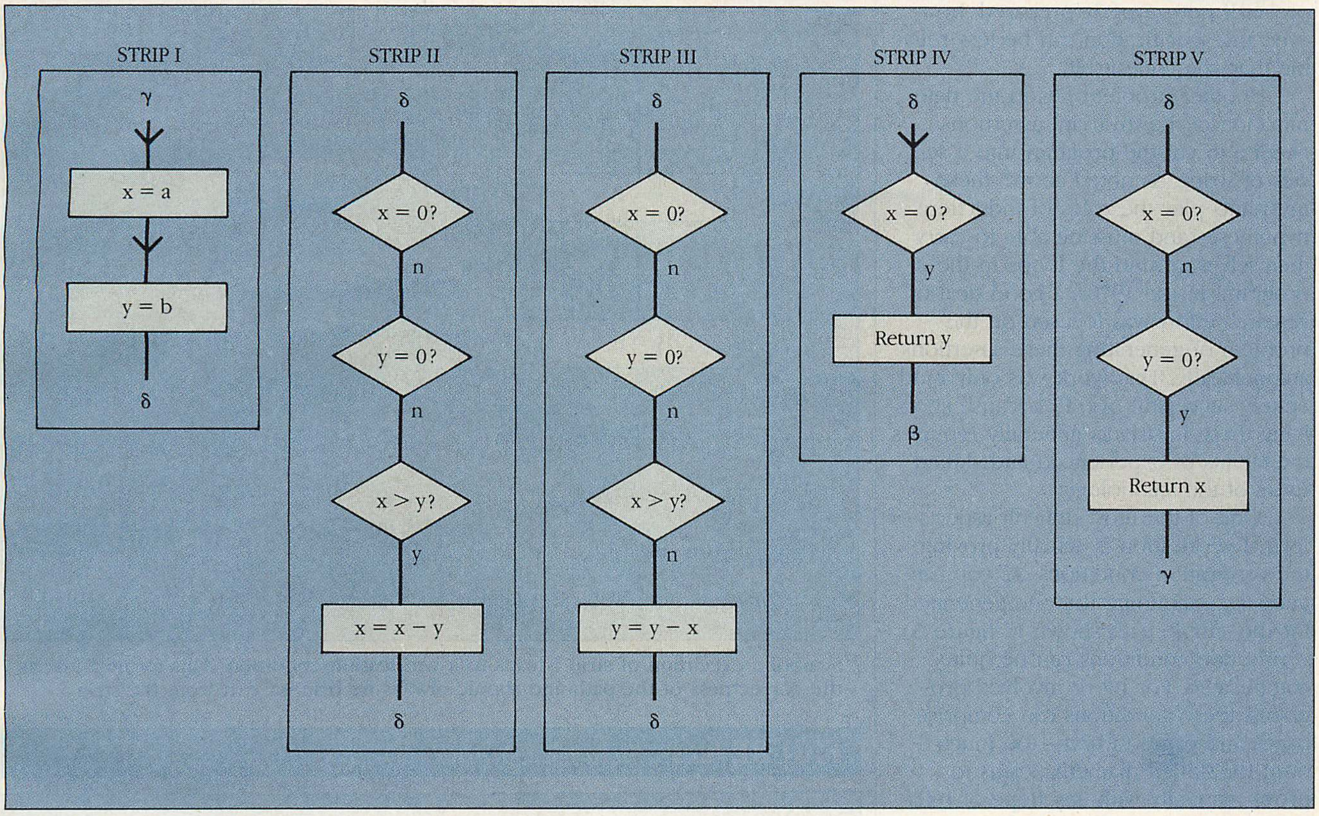
The proof strategy involves considering the strips individually. For each one, it must be proved that if the assertion labeling the top of the strip holds for particular values of the variables, and the strip is executed for those values, then the assertion labeling the bottom of the strip will hold when execution reaches that point. It then follows that any execution path satisfying the input assertion at the beginning of execution necessarily satisfies the output assertion at the end of execution.

To verify each strip, the symbolic execution procedure (explained above) is used to generate a verification condition. The verification condition is then proved, either manually or with the help of an automated theorem prover.

Figures 5 and 6 show the assertions that are obtained from symbolic execution and the resultant verification conditions for strips I and II, respectively. The verification condition for strip I is trivial; however, proving the verification condition for strip II does requires some reasoning in elementary number theory.

Because strip II represents a loop in the flowchart, the same assertion, δ, labels both its top and its bottom. The verification condition for this strip as-

FIGURE 4: *Flowchart Decomposition into Strips*



The flowchart has been decomposed into strips where each strip represents a straight-line path through the program.

serts that the invariant δ is strong enough to carry itself around the loop. If you are familiar with the principle of mathematical induction, think of this invariant as a kind of induction hypothesis. Instead of proving that "if the hypothesis is true for n, it is also true for $n + 1$," we must prove that "if the hypothesis is true for the current values of the variables, it remains true after traversing the loop once."

VERIFICATION DIFFICULTIES

The practical problems encountered when applying formal verification techniques to real-world problems can be formidable. As we have noted, formal methods require that the intended behavior be specified in a precise, mathematical way. Depending on the application, the specification process itself can be an enormous task. Expressing the intended behavior of a compiler, for example, requires encoding the semantics of both the language to be compiled and of the target instruction set. Moreover, the proof of correctness is only as good as the specification; if the specification contains errors, the proof may be worthless. In theory, of course, a specification should be a clear, crisp

statement of intent that someone other than its author can examine and approve. If not properly prepared, however, the specification can be less reliable than the code itself.

Another problem lies in the determination of the invariant assertions needed to cut the program into a series of strips. As noted above, these invariants play the roles of induction hypotheses and must be able to carry themselves around the loops in the program. In the 1970s, a good deal of research effort was focused on the problem of generating these assertions automatically. That work was only moderately successful overall because specifying these invariants generally requires the talents of experienced and skilled users of the technology.

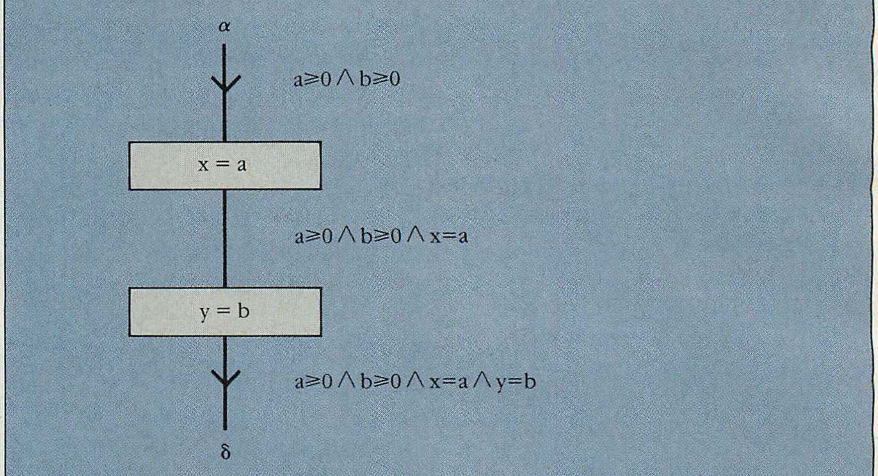
One of the most difficult aspects of formal verification is actually proving the verification conditions. As you can guess by examining the one generated for the simple path shown in figure 6, verification conditions can be quite complicated. For more involved programs, these conditions can comprise pages and pages of text—too much even for skilled mathematicians to prove manually with any hope of reliability. Fortunately, these kinds of formulas are not usually very profound in the mathematical sense, thus they lend themselves favorably to machine-aided reasoning techniques.

In spite of the difficulties, much progress has been made in the last decade in the development of computer-based formal verification systems. The authors of this column, for example, have designed and implemented an experimental specification and verification system that has been used to prove correctness properties of a fault-tolerant distributed computer. More recently, these techniques also have been applied to the verification of microprocessor designs.

Verification of real systems is an extremely time-consuming and expensive proposition. The process costs roughly $1,000 a line (and that cost per line increases as the number of lines increases). Nevertheless, for applications that involve the risk of human life, or whenever extreme reliability is necessary—in other words, when the program simply *must* work—the cost can be well worth it.

Robert E. Shostak, Ph.D., and Richard L. Schwartz, Ph.D., are vice presidents of development and cofounders of Ansa Software.

FIGURE 5: *Verification Condition of Strip I*

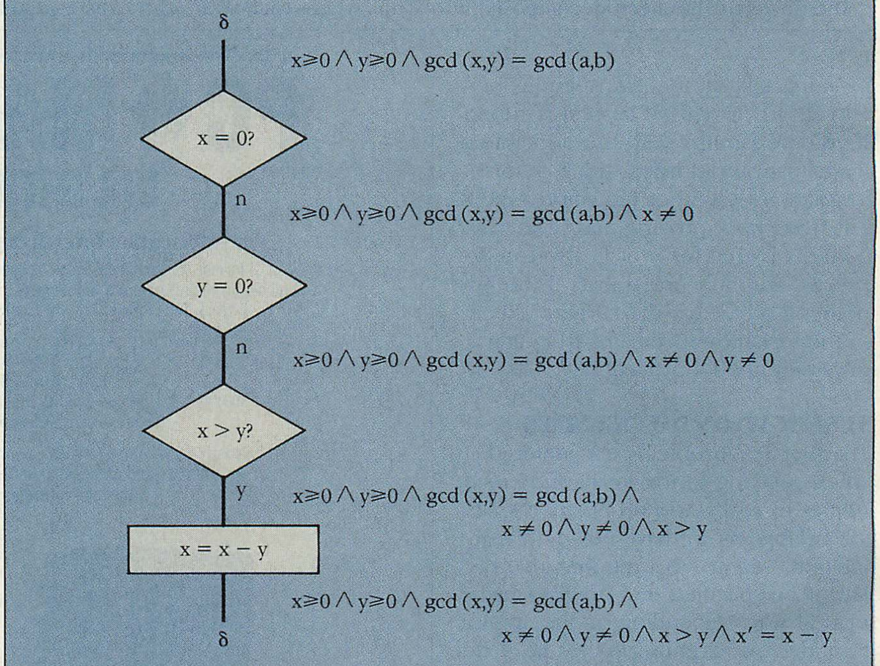


VERIFICATION CONDITION:

$a \geq 0 \wedge b \geq 0 \wedge x=a \wedge y=b \supset x \geq 0 \wedge y \geq 0 \wedge \gcd(x,y)=\gcd(a,b)$

Symbolic execution of strip I leads to a verification condition. This formula asserts the correctness of the path and should always be true on exit from the strip.

FIGURE 6: *Verification Condition of Strip II*



VERIFICATION CONDITION:

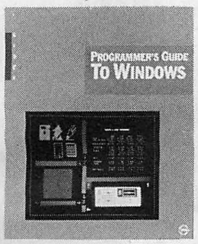
$$x \geq 0 \wedge y \geq 0 \wedge \gcd(x,y) = \gcd(a,b) \wedge x \neq 0 \wedge y \neq 0 \wedge x > y \wedge x' = x - y$$
$$\supset$$
$$x' \geq 0 \wedge y \geq 0 \wedge \gcd(x',y) = \gcd(a,b)$$

Similar to strip I in figure 5, symbolic execution of strip II leads to a verification condition. This formula asserts the correctness of the path.

Windows Opened

With Programmer's Guide to Windows, *Durant, Carlson, and Yao have produced a masterful introduction to Windows programming.*

Programmer's Guide to Windows
David Durant, Geta Carlson, Paul Yao (Sybex, Alameda, CA 1987) 645 pages, paper, $21.95



Despite Microsoft's best efforts in documentation, the daunting steepness of the learning curve for developing a Windows application is quite intimidating. The Windows application development toolkit, although useful, is not a complete teaching device for programming.

The difficulties of learning Windows programming can be traced to the complexity of the environment: its object orientation (subroutines often pass information as objects named for the information they contain), the 350 different subroutine calls, the relatively low-level at which Windows functions, and the messaging system for passing event notification to the program. Windows-programming difficulties also stem from the philosophical style and complexity of the operating environment. This complexity boils down to a simple paradigm: a Windows program becomes a customizing extension to the operating system with the operating system controlling system resources (as opposed to traditionally structured DOS applications where the reverse is true). With all of its intricacies, Windows programming needed a book to rationally explain "how to do it." Enter *Programmer's Guide to Windows*, a sorely needed treatment of the topic, and it comes none too soon.

In this book, Durant, Carlson and Yao introduce Windows programming in a tutorial fashion, taking the time to explain the concepts of the Windows Development environment. The approach is straightforward: the authors take a simple, working shell application and step it through many iterations as the text explores the numerous features of Windows. Because most Windows development is done in C, the authors expect the reader to be familiar with that language.

The opening chapters introduce the Windows philosophy of programming, the terminology, and discuss the basic skeletal structure of a Windows application. Here, the concept of a window is introduced as well as the basic objects that form the Windows environment: pens, brushes, fonts, display contexts, bit maps, and color.

By way of describing the common code present in all standard Windows applications, the Skeleton program is introduced in the third chapter. This program outlines the different initialization steps required of every "good citizen" program operating under Windows. The shell will form the basis of the remaining applications covered.

After covering the foundations: messaging, multitasking, objects, and message translation, the central chapters of the book take the reader through the Windows basics. This topic covers interactive communication with the user using all of the Windows features: scroll bars, child windows, buttons, dialogue boxes, the mouse, cursors, DOS files, and the graphical device interface (GDI).

The book also covers Windows development utilities. By the end of this segment, most readers will be able to produce a functional Windows application. The middle chapters also may stand freely as references. The discussion of the GDI, it is worth pointing out, is one of the best explanations of the topic of viewports and windows available. The treatment of coordinate transformations is also noteworthy.

In its final segments, the book tours through dynamic libraries, memory management, and interapplication data exchange. All of these points are specific to the version of Windows and are appropriately brief. The book concludes with a description of the actual mechanics of producing a Windows application: programming tools and debugging tools. The text describes the key elements of 8086 instruction set in order to understand the Windows debugger, Symdeb. This useful section lists the Windows RIP codes—the codes that Windows displays whenever it detects a fatal error.

Key points about Windows programming style are woven into liberal doses of sample programs. The examples themselves form the building blocks for an individual programmer's own Windows application.

The book is well-written, its explanations are cogent, and the examples are concise. Important, relevant points deserving fuller explanations are placed in sidebars. The figures amplify key points in the text, aiding in comprehension. The index is complete, and the type is very readable.

While a few errors exist, most are insignificant. The description of the operation of the spooler under the GDI is thin. Several program examples introduce topics that are never covered in the text. But, given the complexity of the topic, the book's infrequent out-of-sequence progression does not detract from its overall quality.

The book is compulsory reading for any serious developer of software operating under Windows. Even with the instability of the Windows development platform, as evidenced by the advent of the Presentation Manager under OS/2, the examples presented in this book provide sufficient grounding in Windows programming to allow immediate upgrading to the Presentation Manager when it becomes available. *Programmer's Guide to Windows* should be on every OS/2 and Windows programmers' shelves.

—JOHN COCKERHAM

NEED IT FAST? WANT TO KNOW MORE? DON'T HAVE TIME TO SHOP AROUND?

INTRODUCING...

If you're like most system professionals you're up-to-date about the products in the PC marketplace. You're aware of brand and model differences, are informed about connectivity and compatibility problems, and you shop for competitive prices and fast service.

You're also probably among the many PC TECH JOURNAL readers who purchase by mail. That's why we're starting THE MART—PC TECH JOURNAL'S First Class Mail-Order Section.

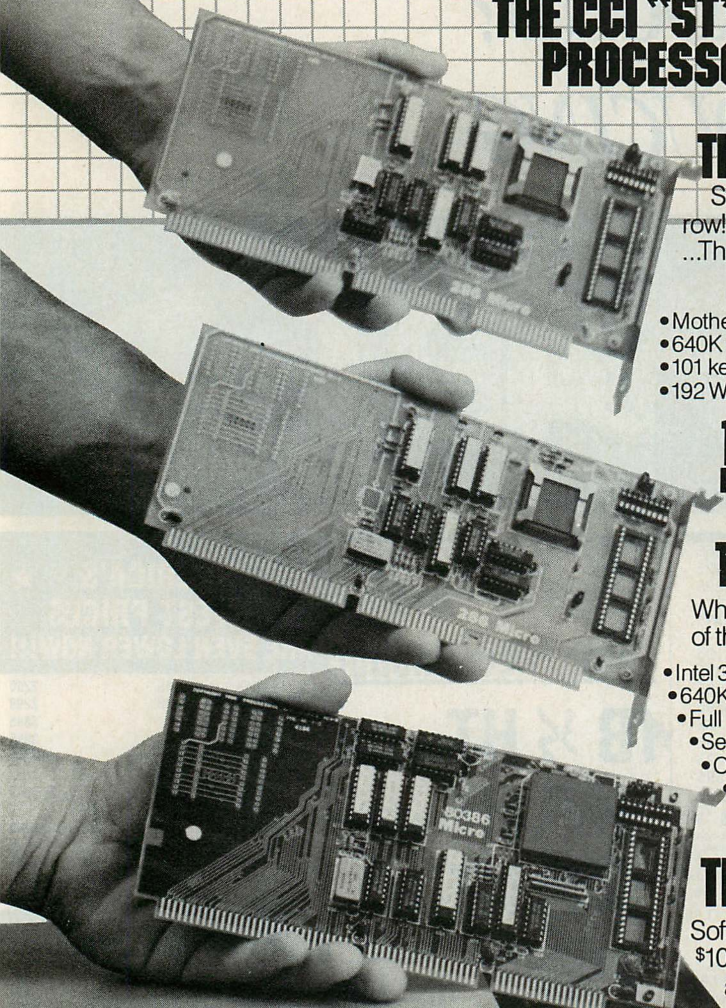
Starting this month, and every month hereafter, you'll find the products you're looking for advertised in THE MART—and you'll benefit from the fast service and helpful support that identifies PC TECH JOURNAL advertisers.

If you're ready for First Class service, you're ready for THE MART.

THE 286/386 ALL PRO TEAM

FULL ONE YEAR WARRANTY
30 DAY MONEY BACK GUARANTEE*

THE CCI "ST" ACCEPTS YOUR CHOICE OF PLUG-IN PROCESSORS FOR COMPLETE UPGRADABILITY.



THE CCI ST/286-10 OR ST/286-12

Stop buying technology that becomes obsolete tomorrow!...CCI finally gives you the power of expandability ...The CCI ST/286 10MHz or CCI ST/286 12MHz 'AT'.

Includes:

- Motherboard with Intel 80286
- 640K of RAM, 1.2 Meg floppy
- 101 key, enhanced keyboard
- 192 Watt power supply
- Clock-calendar with battery back-up
- Enhanced set-up and diagnostic software and all documentation
- Western Digital controller card

80386 Upgrade From $800

10 MHz $1495 **12 MHz $2395**

SEE US AT COMDEX

THE CCI ST/386

When the going gets tough, call in the speed and power of the CCI 80386...**This Powerhouse Includes:**

- Intel 32 bit 80386, at 14.2 MHz
- 640K RAM and 1.2 Meg floppy
- Full 101 key enhanced keyboard
- Serial and parallel ports
- Optional high-speed 80287
- Norton Rating avg. 18.4
- 3 eight-bit, 3 sixteen-bit and 1 thirty-two bit expansion slot

ONLY $3195

THE CCI ST/386 WITH PC-MOS 386

Software Link's PC—MOS 386 offers the speed and power of a $100,000 minicomputer in a desktop PC. It allows multi-tasking, accommodates up to 25 users, and gives them access to up to four gigabytes of memory...***Includes:***

- The CCI ST/386
- 2.6MB of RAM,1.2 Meg floppy
- 101 key, enhanced keyboard
- Serial and parallel port
- PC-MOS 386
- Monographics adapter
- Monochrome Monitor
- A high speed 44MB drive

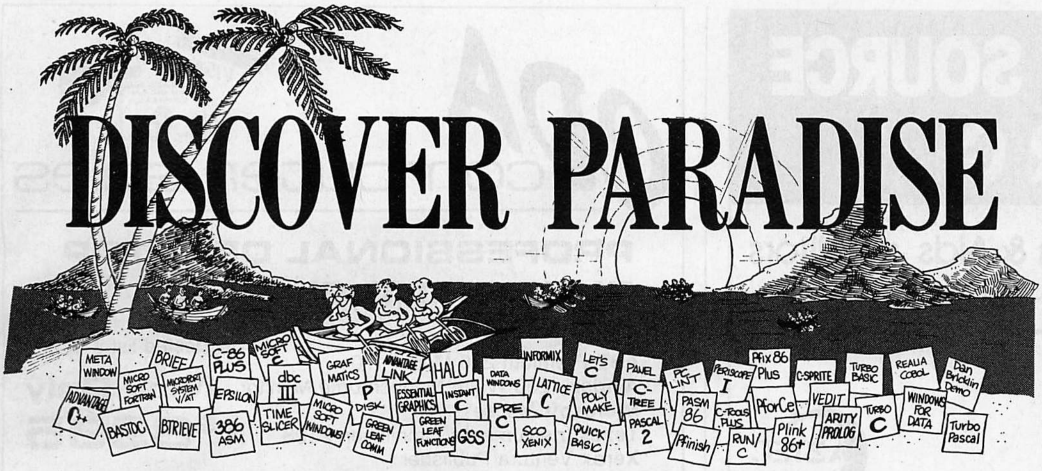
ONLY $4995

CCI *Computer Classifieds, Inc.*
17830 State Road 9, Miami, Florida 33162
(305) 651-5853 Telex 510-600-7725

Call Today (800) 331-5150
In FL Call (305) 651-5853
Support Line (305) 651-0073

No charge for UPS ground shipping. No surcharge for MasterCard or Visa. Florida residents add 5 percent sales tax. ***30-Day Guarantee**-Call or write for details. All Trademarks Acknowledged.

CIRCLE NO. 235 ON READER SERVICE CARD



Programmer's Paradise Gives You Superb Selection, Personal Service and Unbeatable Prices!

Welcome to Paradise. The MS/PC-DOS and XENIX software source that caters to your programming needs.

Discover the Many Advantages of Paradise...

- Lowest price guaranteed
- Latest versions
- Huge inventory, immediate shipment
- Knowledgeable sales staff
- Special orders
- 30-day money-back guarantee

Over 500 brand-name products in stock—if you don't see it, call!

We'll Match Any Nationally Advertised Price.

	LIST	OURS
C++		
ADVANTAGE C++	$ 495	479
PFORCE++	395	225
C COMPILERS		
C86 PLUS	497	369
DATALIGHT C DEVELOPER'S KIT	99	79
DATALIGHT OPTIMUM C	139	105
LATTICE C	500	265
W/SOURCE	900	495
LET'S C	75	55
W/CSD DEBUGGER	150	99
MICROSOFT C V 5.0	450	269
QUICK C	100	69
TURBO C	100	65
C INTERPRETERS		
C-TERP	300	229
INSTANT C	500	379
RUN/C	120	79
RUN/C PROFESSIONAL	250	155
ASSEMBLERS, LINKERS		
ADVANTAGE DISASSEMBLER	295	265
ADVANTAGE LINK	395	359
ASMLIB	149	129
MACRO ASSEMBLER	150	93
PASM-86	195	109
PLINK 86 PLUS	495	299
UNIWARE X-ASMS	CALL	CALL

FREE SCIENTIFIC & ENGINEERING SOLUTIONS CATALOG

Describes 100 of the best software packages for solving complex equations, number crunching, analyzing data, 3-D CAD/CAM design, technical word processing, circuit design, sophisticated graphics and more. Call for your FREE catalog today!

	LIST	OURS
GRAPHICS		
ADVANTAGE GRAPHICS	250	225
ESSENTIAL GRAPHICS	250	185
GRAPHIC	350	285
GSS GRAPHICS DEVELOPMENT TOOLKIT	495	375
HALO—ONE LANGUAGE	250	205
HALO—FIVE MICROSOFT LANG.	595	389
HALO-DPE	195	155
METAWINDOW	195	159
METAWINDOW PLUS	275	229
METAFONTS	95	CALL
METAFONTS PLUS	275	CALL
TURBO WINDOW/C	95	79
C UTILITY LIBRARIES		
ASYNCH MANAGER	175	135
BASIC_C	175	129
BLACKSTAR FUNCTIONS	99	89
C-XPERT	295	249
C FOOD SMORGASBORD	150	95
W/SOURCE	300	179
C TOOLS PLUS	175	135
C TOOLS—TURBO	129	CALL
ESSENTIAL C UTILITY LIBRARY	185	119
ESSENTIAL COMMUNICATIONS	185	125
COMMUNICATIONS PLUS	250	189
GREENLEAF COMM	185	125

	LIST	OURS
GREENLEAF FUNCTIONS	185	125
LINK & LOCATE	350	315
MULTI-C	149	135
PFORCE	295	225
THE HAMMER	195	129
TIMESLICER	295	265
W/SOURCE	1000	CALL
TOPVIEW TOOLBASKET	250	179
SCREEN DISPLAY, WINDOWS FOR C		
CURSES	125	89
W/SOURCE	250	169
FLASH-UP WINDOWS	90	79
GREENLEAF DATA WINDOWS	225	155
W/SOURCE	395	289
MICROSOFT WINDOWS	99	65
WINDOWS DEVELOPMENT KIT	500	309
ON-LINE HELP	149	105
PANEL	295	199
PANEL PLUS	495	395
POLYDESK III	99	75
POLYDESK ADD ONS	CALL	CALL
QUICKSCREEN	195	175
SCREENPLAY (LATTICE)	150	135
SOFTSCREEN HELP	195	169
VIEW MANAGER	275	199
VITAMIN C	225	165
VC SCREEN	99	79
WINDOWS FOR C	250	189
WINDOWS FOR DATA	395	315
Z VIEW	245	169
FILE MANAGEMENT		
BTRIEVE	245	185
XTRIEVE	245	185
REPORT OPTION	145	99
BTRIEVE/N	595	455
XTRIEVE/N	595	455
REPORT OPTION/N	345	269
C-TREE	395	315
R-TREE	295	239
C-TREE/R-TREE BUNDLE	650	519
DATAFLEX	695	649
DBC III/II	250	169
W/SOURCE	500	359
DBC III PLUS	750	595
W/SOURCE	1500	1185
DB_VISTA	195	149
W/SOURCE	495	389
DB_QUERY	195	149
W/SOURCE	495	389
INFORMIX ESQL/C	595	479
INFORMIX 4GL	995	789
INFORMIX SQL	795	639
PHACT	295	265
MAKE, LINT, PROFILE UTILITIES		
ADVANTAGE MAKE	125	99
CODE SIFTER	119	89
LMK	195	139
PC-LINT	139	99
PFINISH	395	219
PMAKER	125	79
POLYMAKE	149	125
PRE-C	295	155
SAPIENS MAKE	79	75
THE PROFILER	125	89
DEBUGGERS		
ADVANCED TRACE 86	175	119
BREAKOUT	125	89
C-SPRITE	175	125

	LIST	OURS
CSD DEBUGGER	75	59
PERISCOPE I	345	285
PERISCOPE II	175	139
PERISCOPE II-X	145	105
PERISCOPE III	995	825
PFIX 86 PLUS	395	229
XVIEW 86	60	49

80386 TOOLS

ADVANTAGE 386 C	895	805
ADVANTAGE 386 PASCAL	895	805
PHARLAP 386 ASM/LINK	495	395
PHARLAP 386 DOS-EXTENDER	995	845
PHARLAP 386 DEBUG	195	155
PC-MOS/386 1 USER	195	179
5 USERS	595	549

EDITORS		
BRIEF	195	CALL
DBRIEF	95	CALL
BRIEF/DBRIEF BUNDLE	285	CALL
CVUE W/SOURCE	250	199
EDIX	195	155
EMACS	295	265
EPSILON	195	149
FIRSTIME (C)	295	229
KEDIT	125	99
LSE	125	99
PC/VI	149	109
PMATE	195	115
SPF/PC	195	145
VEDIT	150	99
VEDIT PLUS	185	129
ARTIFICIAL INTELLIGENCE		
ARITY STANDARD PROLOG	95	79
GOLDEN COMMON LISP	495	CALL
MICROSOFT LISP	250	155
PC SCHEME	95	85
SMALL TALK V	99	85
TURBO PROLOG	100	65
TURBO PROLOG TOOLBOX	100	65
VP EXPERT	100	89
ADDITIONAL PRODUCTS		
COMMAND PLUS	80	69
DAN BRICKLIN'S DEMO PROGRAM	75	59
DISK OPTIMIZER	60	55
DIGITAL RESEARCH PRODUCTS	CALL	CALL
FASTBACK	175	135
MICROSOFT LEARNING DOS	50	39
NORTON COMMANDER	75	55
NORTON UTILITIES	100	59
PDISK	145	89
PFANTASY PACK	995	599
POLYBOOST	80	69
POLYTRON PVCS	CALL	CALL
SIDETALK	120	89
SOURCE PRINT	97	59
TREE DIAGRAMMER	77	65
VENTURA PUBLISHER (XEROX)	895	CALL
VENTURA PUBLISHER (TURBO)	1595	CALL
PASCAL COMPILERS		
MARSHAL PASCAL	189	169
MICROSOFT PASCAL	300	185

	LIST	OURS
PASCAL-2	350	329
TURBO PASCAL	100	65
BORLAND TURBO PASCAL ADD ONS	CALL	CALL
TOOLS FOR TURBO PASCAL		
FLASH-UP *NEW*	89	79
FLASH-UP TOOLBOX	49	45
METRABYTE DATA ACQ. TOOLS	100	89
T-DEBUG PLUS	60	49
TURBO ASYNCH PLUS	100	79
TURBO EXTENDER	85	65
TURBO HALO	129	99
TURBO MAGIC	99	CALL
TURBO OPTIMIZER	75	65
TURBO PROFESSIONAL	70	49
TURBO POWER TOOLS PLUS	100	79
TURBO POWER UTILITIES	95	79
TURBO WINDOW/PASCAL	80	69

NEW PRODUCTS

DATAFLEX—True relational database system that is a powerful, flexible development tool. Includes application generator, relational database manager, on-line multi-file query, forms processing system, on-line multi-key ISAM, macro command language, compiler & text editor.
List: single: $695 — **Ours $629**
multi-user: $1,250 — **Ours $1,149**

C-XPERT—Expert system development tool for incorporating AI programming capabilities into your C programs. Includes backward & forward chaining, atttribute value pairs, frames, inheritance, explanations, why facilities, legal and default values & powerful text features.
List $295 — **Ours $259**

ACTOR—Powerful new language built around object-oriented programming, with windows being defined as objects. Actor makes it easier to include & control windows in application programs.
List: $495 — **Ours CALL**

QUICK C—Microsoft's reply to Turbo C. Fast and complete; compilation speed of 10,000 lines/min. Includes source-level debugger, graphics libraries, library manager, make facility.
List $100 — **Ours $69**

BASIC		
BASTOC	495	399
BETTERBASIC	199	129
BETTERTOOLS	95	89
FINALLY	99	89
MICROSOFT QUICKBASIC	99	65
QUICK PAK	69	59
TRUE BASIC	150	105
TURBO BASIC	100	65
COBOL COMPILERS/UTILITIES		
COBOL SPII	395	329
MICRO FOCUS COBOL	CALL	CALL
MICROSOFT COBOL	700	439
MICROSOFT SORT	195	129
MICRO/SPF	175	155
OPT-TECH SORT	149	115
REALIA CICS	995	785
REALIA COBOL	995	785
REALIA MENU	150	119
RM/COBOL	950	649
RM/COBOL 8X	1250	909
SCREENIO	400	CALL
SCREENPLAY	175	129
VISUAL COBOL (MBP)	795	695
FORTRAN COMPILERS/UTILITIES		
LAHEY FORTRAN	477	CALL
LAHEY PERSONAL FORTRAN	95	CALL
MICROSOFT FORTRAN	450	269
RM FORTRAN	595	399
DIAGRAM'ER	129	115
DOCUMENT'ER	129	115
FORLIB PLUS	70	45
IO/PRO	149	129
MATH PAC	495	445
NO LIMIT	129	115
SSP/PC	350	269
STRINGS AND THINGS	70	49

XENIX/UNIX PRODUCTS

SCO XENIX SYSTEM V-COMPLETE	1295	995
SCO XENIX OPERATING SYSTEMS	595	499
MICROPORT SYSTEM V/AT	549	465
MICROPORT SOFTWARE DEV. SYS.	249	209
OTHER SCO & MICROPORT	CALL	CALL
ADVANTAGE C++	695	CALL
BTRIEVE	595	459
C-TERP	498	379
INFORMIX ESQL/C	749	599
INFORMIX 4GL	1500	1235
INFORMIX SQL	995	795
MICROSOFT LANGUAGES	CALL	CALL
PANEL	625	535
PANEL PLUS	795	675
REAL TOOLS	149	89
RM/COBOL	1250	949
RM/FORTRAN	750	549

OTHER LANGUAGES		
ACTOR	495	CALL
APL*PLUS	595	429
CLIPPER	695	459
JANUS/ADA PACKS	CALL	CALL
LATTICE RPG II COMPILER	750	629
LOGITECH MODULA-2	CALL	CALL
PC FORTH	150	109
PERSONAL REXX	125	99
POCKET APL	95	75

Terms and Policies
- **We honor MC, VISA, AMERICAN EXPRESS**
No surcharge on credit card or C.O.D. Prepayment by check. New York State residents add applicable sales tax. Shipping and handling $3.00 per item, sent UPS ground. Rush service available, prevailing rates.
- Programmer's Paradise will match any current nationally advertised price for the products listed in this ad.
- Prices and Policies subject to change without notice.
- *New* **Extended Hours 9AM EST—7PM EST**

*Ask for details. Some manufacturers will not allow returns once disk seals are broken.

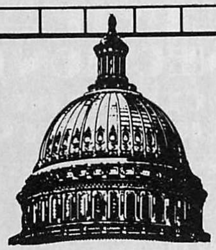
Corporate Buyers—Call for special discounts and benefits!

1-800-445-7899
In NY: 914-332-4548
Customer Service: 914-332-0869
International Orders: 914-332-4548
Telex: 510-601-7602

Programmer's Paradise
42 River Street, Tarrytown, NY 10591

Programmer's Paradise

CIRCLE NO. 173 ON READER SERVICE CARD



"Serving the Nation's Capitol and the World"

Technical Support and Shipping and Receiving
(703) 761-6177, 78
ITS TEK-NET-BBS (703) 690-7462

Open 7 days a week
(703) 847-4740 (800) 642-2395
Information and Technology Services, Inc.
Micro Systems Specialists
8478A Tyco Rd., Vienna, VA 22180

Visa, MC, CHOICE, AE
Leasing, Renting &
Financing available

Software

SUPERCALC 4	$335
SUPERPROJECT +	$295
WORDPERFECT 4.2	$250
DBASE 3 +	$430
FOXBASE +	$335
FRAMEWORK II	$399
RBASE 5000	$320
RBASE SYSTEM V	$396
CROSSTALK XVI	$99
REFLEX	$96
TURBO PASCAL	$42
TURBO PROLOG	$60
TURBO LIGHTNING	$61
NORTON UTILITIES	$57
MS WINDOWS	$69
MS QUICKBASIC	$65
MS C-COMPILER	$298
MS WORD	$280
WORDSTAR 2000	$270
MULTIMATE	

Easy Business Accounting Systems

GENERAL LEDGER	$395
ACCOUNTS RECEIVABLI	$395
ACCOUNTS PAYABLE	$395
INV. CONTROL	$395
RETAIL INVOICING	$395
PAYROLL	$458
ORDER ENTRY	$395
TIME, BILLING & REC.	$635

Video Cards

QUAD EGA +	$379
PARADISE AUTO SWITCH	$419
STB EGA	$299
TECHMAR EGA	$299
VEGA DELUXE	399

NOVELL NETWORKING
DESIGN & INSTALLATION

Drives

20 MB SEAGATE 65MS	$399
30 MB SEAGATE RLL	$499
20 MB ST4026 AT	$573
30 MB ST4038 AT	$651
40 MB ST4051 AT	$792
80 MB ST4096 AT	$1273
40MB PRIAM XT	$1395
60MB PRIAM XT	$1450
40MB PRIAM AT	$1175
60MB PRIAM AT	$1350
BERNOULLI DUAL 10	$1939
BERNOULLI DUAL 20	$2594
20MB PLUS HARDCARD	$695
20MB MAYNARD HCARD	$759
ISI WORM 220MB INT.	$3295
TEAC 360K DRIVE	$89
60MB ARCHIVE TAPE	$740
60MB ARCH. EXT. TAPE	$740
60MB GENOA TAPE	$935
60MB SYSGEN	$1089
TOSHIBA 3.5 DRIVE	$150
TOSHIBA 10 MB DRIVE	$899
TOSHIBA 5.25 EXT.	$349

Memory Boards

JRAM 2	$129
JRAM 3 ABOVEBOARD	$179
JRAM AT	$179
JRAM AT3 ABOVEBOARD	$239
JLASER MODULE	$265
JLASER +	AVAIL.
INTEL ABOVEBOARD	$438
AST RAMPAGE 512K	$485
ORCHID CRAMRAM	$291
ZUCKER BOARD	$68
AST 6 PAK + W/384	$199
QUADBOARD W/384	$189

Chips

64K 150ns set of 9	$8
64K 120ns set of 9	$9
256K/150ns set of 9	$23
256K/120ns set of 9	$25
64 x 4	$4
128K Piggyback	$5
8087-3	$119
8087-8	$169
80287 5MHz	$239
80287 6MHz	$245
80287 8MHz	$318

Printers

NEC P-6	$480
NEC P-7	$685
NEC P-5	$1122
NEC P-5XL	$1245
PANASONIC 1080 I	$235
PANASONIC 1091 I	$299
PANASONIC 1092	$360
PANASONIC 1592	$480
STAR LV1210	$189
STAR NX-15	$383
STAR ND-15	$445
STAR NR-15	$537
STAR NB 24-15	$699
STAR SD-10	$355
STAR NX-10	$255
BROTHER 1509	$440
CITIZEN MSP-10	$320
CITIZEN MSP-15	$435
CITIZEN PREMIER 35	$485
TOSHIBA P341	$835
TOSHIBA P351	$1227
FUJITSU DLP24	$1239
FUJITSU DM9I	$413
OKIDATA M182	$253
OKIDATA M192 +	$412
OKIDATA M193 +	$613
OKIDATA M84	$713
OKIDATA 2410	$1976
EPSON FX-286	$624
EPSON LX-80	$285
EPSON FX-85	$434
EPSON LQ-800	$631

Laser Printers

CANON A1	$2095
CANAON A2	$3085
HP LASERJET	$2295
QMS KISS	$1995
XEROX 4045	$CALL

Specials

Color RGB Mon.	$305
Amdek 310A	$150
TEAC 360 Drive	$89
Samsung Monitor	$89
Amdek 722	$549
NEC Multi-Sync	$631

**Other Systems

Sharp PC 7000	$1095
IBM XT	$1798
IBM AT	$2789
BIOS AT 8MHZ	$1769

- Prices subject to change 12/18/86
- 10% re-stock fee on all items
- Software non-returnable if opened
- No surcharge on VISA, MC, CHOICE. . . AE, 3%

'Editor's Choice"

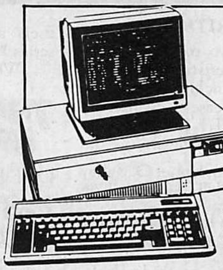
ITS Turbo XT

1 year warranty
30 day money back guarantee

The $895 ITS Turbo XT from Information and Technology Services is our "best buy" recommendation with 20MB $1224.00

PC MAGAZINE
OCT. 14, 1986
IBM® PC Compatible

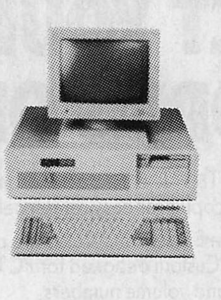
$895.00 complete



SPERRY PC/IT

Superior to the IBM AT
Rated 8.8 by InfoWorld
44MB, 28ms Access Hard Drive
3 Speed Processor
Free 8MHz Math Coprocessor
1.2MB Floppy , 640K Ram
2 Serial/1 Parallel Ports
Clock/Calendar, AT Keyboard
DOS 3.1, Basic, System Guide

$2,895.00 1 year warranty



ITS SYSTEM 386 *BLACKHAWK*

Features:
- INTEL 80386 CPU and support circuits
- INTEL designed motherboard
- Phoenix BIOS
- 18 Mhz clock speed
- PC/AT compatible 8 Mhz switchable from keyboard
- 512K RAM standard up to 14 megabytes
- Parallel/Serial/Clock
- 8 Slot Expansion bus interface
- 2- PC Compatible 8 bit bus connectors
- 2- 32 bit bus connectors
- Hard disk/Floppy disk controller
- 1.2 megabyte floppy

Available now
$4495.00

TOSHIBA Lap-Top T-1100 Plus

(DUAL FLOPPY)

10 MB Disk Drive Now Avail.

$1695.00

$1995 with 640K

- 256K Ram Memory • CMOS 80C86 Running at 7.1 MHz • Two 720K 3.5" Floppy Drives
- One Expansion Slot • TOSHIBA Keyboard • LCD Display
- Color Graphics/Monochrome Composite Card • One Parallel and Serial Port • Clock Calendar
- External Floppy Drives Optional • DOS 2.1

1800 +

80286 CPU

- IBM AT Compatible
- 512K of RAM expandable to 1MB
- 1.2MB Floppy
- AT Keyboard
- Documentation and Diagnostics
- Made in the USA

$1195.00

CIRCLE NO. 134 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PRODUCT CATEGORIES

Advertising Rates and Information: PC Tech Journal Marketplace

PC Tech Journal Marketplace is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches. Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $170 per issue ($510 total).

For additional information call 212-503-5115.

PC Tech Journal Classified Advertising Staff
One Park Avenue, New York, NY 10016
(212) 503-5115

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Daniel L. Rosensweig

Sr. Advertising Coordinator
Monica Dixon

Advertising Coordinator
Angela Kiffin

Sales Assistant
Linda Annis

Production Manager
Anne R. Brockinton

Production Coordinator
Elliot Appel

Account Managers

Stanley H. Robinson (212) 503-5116
AL, AR, IA, IL, IN, KS, KY, AZ, CO, OR, NM, LA, MI, MN, MO, MS, NB, ND, OH, OK, SD, TN, TX, NV, AK, GA, UT, CA (ZIP 92999 & DOWN), CANADA (OTHER THAN BRITISH COL.) AND ALL OVERSEAS CALLS.

Lisa B. Stick (212) 503-5172
CT, MA, ME, NH, NJ, NY, RI, ID, MT, MD, VT, DC, DE, HI, NC, SC, FL, VA, WV, WI, PA, WA, WY, CA (ZIP 93000 & UP) BRITISH COL.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

HARDWARE/ACCESSORY CARDS

DIGITAL SIGNAL PROCESSOR
The *Model 10* coprocessor board is based on the 16/32 bit TI TMS 32010 and is designed for applications in communications, speech, instrumentation, and numeric processing. A 1K complex FFT takes 90ms. Offered with onboard 12 bit 80 Khz A/D and D/A. Includes all utility and applications software. $650-$850.
DALANCO SPRY
Suite 241 2900 Connecticut Ave. NW
Washington, DC 20008
(202) 232-7999

ADD 4 SLOTS TO IBM-PC!
The Addcard allows updating your IBM-PC system by adding 4 more card slots. It easily installs in the 5 slot motherboard providing 8 slots total. Accepts many boards including turbo, modem, floppy or hard disk controllers ram, serial-parallel ports, security & others. Only $79.00 + $3 P&H. Visa & M/C accepted.
Merak Industries
8704 Edna
Warren, MI 48093
1-800-231-4310 Ext 768 or 313-562-9768

TECH MARKETPLACE . . .
Second Color Option Available

PC TECH JOURNAL MARKETPLACE is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches.

Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $170 per issue ($510 total).

COMMUNICATIONS Category
begins on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

COMMUNICATIONS

IBM COMPATIBLE RS232 EASI-DISK 3½"/5¼" FLOPPY DATA STORAGE & TRANSFER SYSTEMS

Information Transfer to/from Non IBM Compatible Systems to/from IBM and Compatibles: (Over RS232 Interface or 488 Interface).

- Reads & Writes IBM PC/MS DOS Disks
- RS-232CI/O/488
- Rugged Portable Package/Battery Option
- ASCII or Full Binary Operation
- Baud Rates 110 to 38.4 K Baud
- Automatic Data Verification
- Price $895 in Singles - OEM Qtys. Less.

28 other systems with storage from 100K to 35 megabytes.

ADPI
ANALOG & DIGITAL PERIPHERALS, INC.
815 Diana Drive
Troy, Ohio 45373
513 339-2241
TWX 810 450-2685

CIRCLE 268 ON READER SERVICE CARD

ARC - MAP
Asynch RS-232 Comm - Monitor/Analyzer Prog
Use PC to diagnose data communications. With cables, manual; 2-19200 BPS; 32K bfr; data, errors, mdm sig's; Passive, Active, BERT; ASCII, EBCDIC, Hex, Dec, Oct; bfr search; event traps; timer; save strings, setups, data; EASY use!! $249, 30 day M/B
George Driver
1316 El Rancho Dr.
Bakersfield, CA 93304
(805) 835-1916

COMPATIBLES

AT $1,150 **$1,800 w SEAGATE ST-251 40MB**
HI-RES (720x350) AMBER MONITOR • INTEL 80286-6/10 MHz • ONE MB RAM ON BOARD • PHOENIX BIOS • HERCULES EMULATION GRAPHICS CARD w/PRINTER PORT • SERIAL PORT • OPERATES MS/PC-DOS, XENIX, UNIX, GW BASIC • 8 SLOTS • WD CONTROLLER 2FD/2HD • ENHANCED KEYBOARD w/LEDS • 200 W POWER • ON BOARD CLOCK/CALENDAR • 1.2MB FLOPPY • HARDWARE RESET • SPEAKER • 80287 SOCKET • ACCEPTS ALL IBM PARTS • OPT'L EGA/COLOR "0" WAIT/12 MHz • FULLY SET UP/TESTED • MANUALS • ONE-YEAR PARTS/LABOR

XT $650 **$900 w SEAGATE 225 20MB & 1 FD**
HI-RES (720x350) AMBER MONITOR • INTEL 8068-2 4.77/80MHz OPTIONAL MEC V-20 OR 10MHz RUNNING 300% FASTER THAN IBM PC • 640K ON BOARD • PHOENIX BIOS • 150-W • HERCULES EMULATION GRAPHICS CARD/w PRINTER PORT • OPERATES MS/PC-DOS, GW BASIC • 8 SLOTS • MULTI I/O CARD WITH 2 FD CONTROLLER PRINTER/GAME/SERIAL PORTS • BATTERY BACKUPED TIME/CALENDAR • ENHANCED KEYBOARD/w LEDS • 150 W POWER • TWO 360K FUJITSU DRIVES • 8087 SOCKET • SPEAKER • ACCEPTS ALL IBM PARTS • FULLY SET-UP/TESTED • MANUALS • ONE-YEAR PARTS/LABOR •

386'S & MULTI-USER SYSTEMS PLEASE CALL
PORTABLE XT/AT SYSTEMS TOO
PCPRIME SYSTEMS, INC.
800-451-5279

OPEN 7 DAYS **VISA/MC/AE**

135 W. 26TH ST. 8TH FL.
NEW YORK, NEW YORK 10001
(212) 627-4485

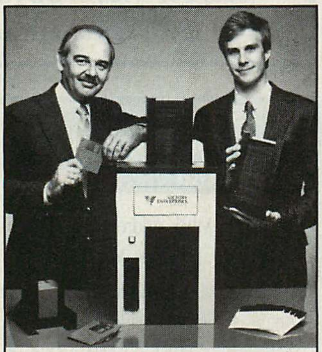
6951 WARNER ST. SUITE 294
HUNTINGTON BEACH, CA 90706
(800) 451-5279

CIRCLE 269 ON READER SERVICE CARD

SOFTWARE/
COMMUNICATIONS
continued on next page

PERIPHERALS

CREATE A DISKLESS PC!
PC-ROMDRIVE allows users to create a "Diskless PC" capable of booting a ROM-resident copy of MS-DOS and/or user application programs. PC-ROMDRIVE consists of a PC-compatible ROM/PROM expansion board and the PC-ROMDRIVE software. PC-ROMDRIVE is priced at $195 for single units. Quantity discounts and OEM arrangements available. MC/VISA
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, Az. 85069
(602) 866-1786



AUTOMATIC DISKETTE DUPLICATOR

- Runs stand-alone or attached to PC
- Copies 5¼-inch and 3½-inch formats, including IBM PS/2
- One-button unattended operation
- Removable input/output cannisters
- Serialize/Print Labels/Test Alignment

$5,995

VICTORY ENTERPRISES
Technology, Inc
Call 1-800-421-0103
(512-450-0801)

CIRCLE 270 ON READER SERVICE CARD

PERIPHERALS

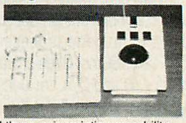
SPEECH SYNTHESIS
SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
55 Park St., Suite 2
Troy, MI 48087
(313) 588-7370

FastTRAP™

The pointing device of the future is here!



- Two and three axis pointing capability
- High resolution trackball for X and Y axis input
- High resolution fingerwheel for Z axis input
- Use with IBM® PC's, XT's, AT's and compatibles
- Three input buttons
- Full hardware emulation of Microsoft® Mouse
- Standard RS-232 serial interface
- Includes graphics drivers and menu generator
- Easy installation
- 1 year warranty
- Made in the U.S.A.

ONLY $149.00
VISA and MasterCard accepted

LTS/C Corp.
319 South Limestone Street
Lexington, Kentucky 40508
(606) 233-4156
(800) 872-7279

3-D TRACKBALL FOR IBM
CIRCLE 271 ON READER SERVICE CARD

SOFTWARE

BUSINESS

LP88—SPREADSHEET LP
Our best-selling linear programming system solves problems w/1000 constraints and 5000 variables, reads/writes Lotus worksheets, many other advanced features. IE News says "the flexibility and features of this program are a bargain at its low price." Req. 192k. $149 w/manual and 8087 support. $29 demo.
Eastern Software Products, Inc.
P.O. Box 15328
Alexandria, VA 22309
(703) 360-7600

COMMUNICATIONS

PC SERIAL DATA ANALYZER
Use your IBM PC or compatible to analyze data streams between serial devices. Windows show each devices transmissions in ASCII or HEX. 64K buffer, Signal line monitoring, disk save, "Lotus" style interface, an invaluable tool for debugging serial interfaces. DISK and MANUAL $150.00
TRIPLE C SOFTWARE
1827 S.W. 24th Ave.
Fort Lauderdale, FL 33312
(305) 583-0687

timeslips™

$99.95

POP-UP TIME & BILLING SOFTWARE

"Jewel of a Billing Program"—INFOWORLD, RATED 9.3

Perfect for Attorneys, Accountants, Consultants PR/Advertising and ANY service professional.

- Track time like a "stopwatch"
- Batch enter time/expense data
- Over 5,000 professional bill formats with aged receivables, flat fee and more
- Produce custom business and financial reports
- Export to spreadsheets and databases
- Track retainer/trust account
- Optional G/L interface now available

NEW!!! **TIMESLIPS III more reports, more capacity, more speed, auto back-up and much more.—$199.95. Hard Disk & 384K required.**

For rush service/more information, call (617) 468-7358.
Add $7 s/h. **Not copy protected. 30-day money back guarantee.**

Call (800) 225-5669 to order

NORTH EDGE SOFTWARE CORP.
P.O. Box 286
Hamilton, MA 01936

CIRCLE 272 ON READER SERVICE CARD

TECH MARKETPLACE . . .
the comprehensive guide to products and services in the MS DOS market.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

COMMUNICATIONS

ETHERNET FOR PCs.

PC/TCP is a complete ethernet package for PCs. Connects PCs to other PCs or UNIX hosts.

Runs on any PC or compatible and requires a 3COM, Interlan, Proteon, or BICC board.

Price: $400 per PC.

UniPressSoftware

2025 Lincoln Hwy.
Edison, NJ 08817
800-222-0550 (Outside NJ)
201-985-8000
Telex: 709418

CIRCLE 273 ON READER SERVICE CARD

ENGINEERING

METAL FABRICATORS

PC/Cultist takes input from your bill of material—Detail drawing and calculates the best cutting combination for any length stock and prints a shop ready cutting list and scrap report. Also an optimization feature finds best multi length for mill orders. Price $300.
Demo Disk $25.00
THE JOSEPH ALBERT CO.
P.O. Box 611
Blue Island, Illinois 60406
(312) 349-9032

ENGINEERING SOFTWARE

You will save hours of development time with our library of integrated programs.

- Graph Plotting
- Equation Processing
- Simultaneous Equations
- Complex Arithmetic
- Pop Up Calculator
- Fourier Transforms
- Top Down Designer
- and more...........

For a FREE CATALOG: write Pulse Research, Dept Eng 4, P.O. Box 696, Shelburne, Vt 05482 (802) 985-2928

CIRCLE 274 ON READER SERVICE CARD

GENERAL

IMPORTANT NEWS FOR DESQVIEW/AST SPECIAL EDITION USERS

Time is running out for you to upgrade to DESQview 2.0 at the special low price. If you act before September 30, 1987 you get DESQview 2.0 for only $50 plus $5 shipping and handling (add 6.5% sales tax in California). After that date you can still upgrade but your cost will be $80 plus $5 shipping and handling. So hurry and save. Send your DESQview registration card and check to:

DESQview 2.0 Upgrade Offer

Quarterdeck Office Systems • 150 Pico Blvd. • Santa Monica, CA 90405
Tel. (213) 392-9851

TAPE/DISK CONVERSIONS

Conversion services to or from over 800 computer systems:
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
PIVAR COMPUTING SERVICES, INC.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

GRAPHICS

COULD YOU PRINT THIS PHOTO

WITH YOUR PROGRAMMER'S EDITOR?

You can NOW with SLEd!

SLEd offers a remarkably simple method of creating graphics and incorporating them into your documents using your editor and a laser printer . . . $149.95

VS SOFTWARE
P.O. Box 6158
Little Rock, AR 72216
501-376-2083

CIRCLE 275 ON READER SERVICE CARD

GRAPHICS

FORTRAN GRAPHICS LIBRARY

GRAFMATIC (screen graphics): 75 MS FORTRAN/Pascal, R-M/Profort, Lahey FORTRAN callable subroutines. Fully documented, prof. graphics capabilities, inc. general utility, 2-D interactive, total 2-D plots, 3-D plots and solid models. $135. H-P or H-I plotter? get PLOTMATIC, complete plotter graphics library. Interfaces w/GRAFMATIC. $135. Both $240.
MICROCOMPATIBLES, INC.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Use high resolution color slides up to 4000 line.
COMPUTER SLIDE EXPRESS $9/slide.
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

C GRAPHICS. ONLY $89!

GRAF-PAC: Over 90 fast functions.
- Complete window management
- Point, line, lineto, polyline
- Circle, ellipse, blockfill, blocksave
- Fonts and font editor
- And much more . . . Supports lattice, MS, Turbo C

The Enaar Software House
P.O. Box 10072
Costa Mesa, CA 92627
(714) 631-2140

PC PEN-PLOTTER EMULATION

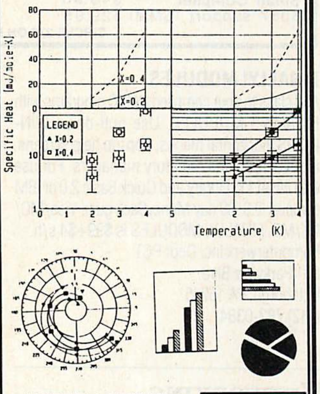
FORTRAN callable, Versatec/Calcomp compatible (PLOTS, PLOT, AXIS, LINE, SYMBOL, etc. plus extras) plotter emulation for CGA, EGA, Herc, Epson, C. Itoh and compatibles. Libraries for MS 3.3/4.0 and Lahey F77L. Includes 100+ page manual, complete examples on disk. Only $150, call for educational discount and PGC prices.
F and S Software
7604 Peacock Drive
Huntsville, Alabama 35802
(205) 881-6268

GRAPHICS

TEMPLE
DATATAP™ GRAPH^C
for
IBM XT,AT,PS/2

Developed at Temple University

BEST SCIENTIFIC PLOTTING SOFTWARE



INTERACTIVE, WYSIWYG TEXT CHARTS TRANSPARENCIES

Introductory Price **$99.00**

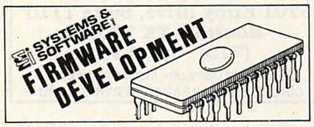
MIHALISIN ASSOCIATES
600 HONEY RUN ROAD
AMBLER, PA 19002
215-646-3814

CIRCLE 276 ON READER SERVICE CARD

LANGUAGES

FIRMWARE DEVELOPMENT IN C

REX-C/86 C package supports ROM code generation. Includes XC86 C compiler which implements draft ANSCI C standard, supports-in-line assembly, produces optimized 8086/87/186 reentrant code for real-time environment, generates separate segment for initialized data and string constants for ROM-based applications, produces object file in Intel OMF with debug information, global and local symbols with data type and line numbers for high-level debugging. Price is $750 which includes XC86, linker, locator, librarian, hex formatters, and run-time library source.



SYSTEMS & SOFTWARE, INC.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

LANGUAGES

** Introductory Special **

FTL MODULA-2 ONLY $79.95

Large Memory Model Modula-2 Compiler for MSDOS Systems. Includes: Compiler, Linker, Editor, 8087 support, and full Library sources. FTL Modula-2 can create programs as large as your available memory.

Don't Delay Order TODAY!!

SAVE $10.00 when you order one or more of the support packages.
Editor Toolkit $39.95
Small Compiler $49.95
8087 support (SMM) $29.95

Workman & Associates
1925 E. Mountain Street
Pasadena, CA 91104
(818) 791-7979

CIRCLE 277 ON READER SERVICE CARD

FINALLY! MODULES

Add class to your compiled BASIC programs with FINALLY ! MODULES. Use pull-down WINDOWS, horizontal menus, pop-up help screens, input screen and directory managers. For use with FINALLY! Library and Quick Basic 2.0 or IBM compiler 2.0. 30 day MoneyBack guar. Visa/MC/CK/MO. FINALLY! MODULES is $99+$4 s/h.
Komputerwerk Inc. Dept PCT
851 Parkview Blvd.
Pittsburgh, PA 15215
(412) 782-0384

<>farbware Modula-2

- Full 3rd Edition Language
- Large Memory Model/Unix make
- DOS Linker Compatible Object
- Both 8087 and 8087 simulation

Very fast compiler, runtime source, many extensions, site licensing, assembler interface. $89.95 MC/VISA/Corp PO/CK
farbware
1329 Gregory
Wilmette, IL 60091
(312) 251-5310

NETWORKING

NOVELL NETWORKING PRODUCTS

When reliability and performance really count you can't afford anything less than Novell. We offer quality Novell products and accessories, fast courteous service and competitive pricing for all your networking needs. Our experienced staff is waiting to serve you. Call today for information and quotes.
BENEDEK ENTERPRISES
104 Cool Springs Road
White Oak, PA 15131
412-751-8381

MENUET™

MULTIUSER LAN MENU SYSTEM

Organize all applications into one menu system
Create professional network menus without batch files
Access programs quickly with one-key or cursor-select
Customize menus, colors, passwords for each user
Give or deny access to DOS to each user
Allow user logon and logoff without reboot
Provide context-sensitive help for menus and options
Run programs, DOS functions, batch files
Create, delete, and modify menus quickly and easily
Password protect user-ID's and menu options
Maintain a virtually unlimited number of user-ID's
Build systems with up to 200 menu screens
Add up to 24 options per menu
Nest menus up to 20 deep
Written in fast, compact 'C' language

$99.50
License for one Network

Network Support Corporation
3701 Kirby Drive, Suite 1110
Houston, TX 77098
(713) 521-1211
Dealer and Distributor Inquiries Welcome
CIRCLE 278 ON READER SERVICE CARD

PROGRAMMERS TOOLS
category continued on next page

OPERATING SYSTEMS

Multitasking
DOS 2.1 - 3.2
PC & 286, 386 real

1) Do windows & use numeric variables in BAT files
2) Run up to 23 programs concurrently with full intertask communications
3) Has a system status and utilities pop-up screen

"Extended DOS" $ 39
Pgm interface tech ref $120

Distributed Software
3951 Emerson St.
Marietta, Ga. 30062
404-977-0152
CIRCLE 279 ON READER SERVICE CARD

PROGRAMMERS TOOLS

IBM PC X-ASSEMBLERS

Develop MPU products on your PC! 21 different fast cross-assemblers. Both absolute and relocatable. Linker, Locater, Librarian, Format Converter, X-reference, Macros, Public, Local symbols and more.

relms

RELMS™ PO BOX 6719
SAN JOSE, CA. 95150
TWX 910-379-0014
(408) 265-5411
(800) 448-4880
CIRCLE 280 ON READER SERVICE CARD

PROGRAMMERS TOOLS

TURBO DEVELOPMENT SYSTEM

Only $39 - A complete environment for developing Turbo Pascal programs. Find and correct all syntax errors in one pass. In many cases you have the option of checking only sections of your program. Includes built-in editor, DOS Shell as well as Delete, Rename and copy without exiting program. Not copy protected. MC & Visa welcome. Add $3 for S/H.
MicroHelp, Inc.
2220 Carlyle Drive
Marietta, GA 30062
(800) 922-3383 in GA (404) 973-9272

TURBO PASCAL GENERATOR

GTP APPLICATION DEVELOPMENT SYSTEM
Builds complete, working applications.
You give it spec's, it writes error-free code

- Indexed Data Bases
- Multiple Screens
- Report Generator
- Menu Generator
- Context-Sens. Help
- Global Searching

Easy to Use Price $200.00 Visa/MC,ck,MO
AEF
P.O. Box 928
Katy, TX 77492
(713) 391-8570

TURBO PACKAGE 3.0 (for Turbo Pascal)

MULTI-TASKING, MULTI-PROCESSING, & MODULAR REENTRANT 640K PROGRAMS

FREE: A multi-processor data entry program for 2 or more machines.

Don't pay high prices for a network. Turbo Package has true multi-processing. Use all the power and memory you need.

A file processor LAN only gives you shared disk drives and a print spooler. Turbo Package lets you run data entry on several PC's, number crunching on a 386, intelligent file processing on an AT/XT, and print spooling. Hardware required: one PC or compatible. Any MIMDA configuration is possible.

Balance your work load and reduce hardware costs. Network with a modem (Wide Area Network), RS232, or RS422 (Local Area Network).

Few code changes required. Call 640K programs, single processor tasks, and multiprocessor tasks just like procedures and functions.

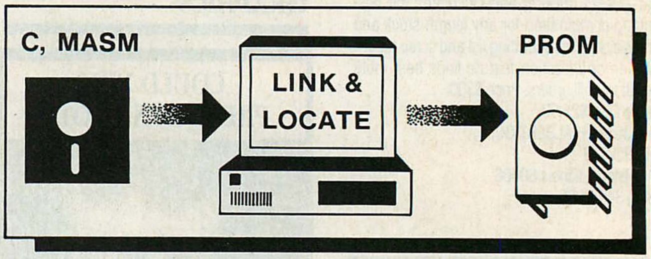
You also get ZapIO, a high speed screen IO that works with writeln().

Turbo Package 2.0
w/ 640k Pascal programs......$59.95
w/ Multi-Tasking..............89.95
Turbo Package 3.0
w/ Multi-Processing..........149.95
SuperMath 32-bit toolbox
40+ math routines............24.95

90 day money back guarantee (In Texas add TAX)
Visa, M/C, or COD Shipping: USA $5
Foreign $10

Conversational Computer Systems
5371 Verbena Rd.
San Antonio, TX 78240
Ph: (512) 692-0353
CIRCLE 281 ON READER SERVICE CARD

FIRMWARE DEVELOPMENT



LINK & LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by popular C compilers, such as from Microsoft, Lattice and Borland's Turbo C, and MASM from Microsoft, Provides full control of segment placement anywhere in memory. Supports output of Intel HEX file for PROM programmers, Intel OMF absolute object file for symbolic debuggers and in-circuit emulators. Includes Intel compatible linker, locator, librarian, hex formatter and cross reference generator. $350.

NEW! Includes utility to support PC based PROM programmers.

SYSTEMS & SOFTWARE INC.
3303 Harbor Blvd., C11, Costa Mesa, CA 92626
Phone (714) 241-8650 FAX (714) 241-0377 TWX 910-695-0125
CIRCLE 282 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

Structure for Assembly?

Unique program adds structure to assembly language. IBM/MASM compatible. Develop and debug faster with structured concepts. If-then-else, select-when-other, do while, do until, do incremental, all combinations, and do forever. Leave/leaveall loop exits, and file includes. Nest 16 deep. $50.

LANEY SYSTEMS, INC.
3 Office Park Drive Ste. 100
Little Rock, AR 72211
501-225-7755

ASSEMBLY LANGUAGE TOOL

VersiCREF™ creates a sorted Master Cross-Reference Listing of an entire application from the MASM CRF files. Pinpoints which source modules are affected by changes to common PROC's, variables, etc. Handles 100+ separate program source files. Full X-Ref or just PUBLIC symbols. $55 MC/VISA/COD/AMEX.

SUMMIT INFORMATION SYSTEMS INC.
73 East Lane
Willingboro, NJ 08046
1-800-334-4096 (in NJ 609-871-0202)

FORTRAN Developers

Essentials for documenting and debugging large FORTRAN programs. DOCUMENT'er: prepares x-references and symbol tables for an ENTIRE program with ALL routine, COMMON, ... definitions + ALL uses. DIAGRAM'er: draws clear diagram(s) of complex code and data structures of your code. Interactive + many display options. Needs only existing source, MAP, and library files. All compilers supported. Both $128.75

IMPULSE
ENGINEERING

Ray Strong, (415) 788-4611
IMPULSE Engineering
P.O. Box 3540
San Francisco, CA 94119-3540

PASCAL-to-C TRANSLATOR

Industrial strength conversion from Turbo, Microsoft, UCSD, MT+, Apollo, Macintosh, and other Pascals to K&R C. Handles nested procedures, intrinsic functions, separately compiled units and modules, all data types including long integers.

Requires 512K IBM PC/XT/AT. Send up to 500 lines of Pascal and we will convert it for FREE. Site licensing from $5,000. Conversions 50 cents/line.

TGL Inc.
27096 Forest Springs Ln.
Corvallis, OR 97330
(503) 745-7476

CROSS-TRANSLATION UTILITIES

Convert 8085, Z80 Assembly language programs to 8086 assembly language programs. 2500 lines converted and re-assembled on IBM PC in 20 minutes. With a error-free listing. Also, 8080/8085 to Z80 or NSC800 conversion. Use Microsoft MASM! Ask for what you need!

relms
RELATIONAL MEMORY SYSTEMS, INC.

RELMS™
PO Box 6719
SAN JOSE, CA. 95150
Call (800) 448-4880 (408) 265-5411
TWX 910-379-0014

BASIC/COBOL SCREEN CODE

- Create screen code for Quick/Turbo BASIC
- ($39), IBM/MICROSOFT/REALIA COBOL ($59).
- Powerful editor. Paint, edit, draw etc. All
- attributes. Foreign language capability.
- REALIA/BASIC includes cursor, border, beep
- etc. COBWORK generates report headers ($39).
- shp $5.

Tajeva Software
6064 Belle Grove Cove S.
Memphis, TN 38115
(901) 365-4692

VERSION CONTROL SYSTEM

****** NEW! TLIB™ 4.0! ******

See review of TLIB V3 elsewhere in this issue of PC Tech Journal. The fastest, most powerful source code control system is now even faster, with many new features: Keyword substitution, generalized wildcard and list-of-file support, branching, more. DOS 2.X/3.X $99.95 $3 s/h.

BURTON SYSTEMS SOFTWARE
P.O. Box 4156
Cary, NC 27519-4156
(919) 469-3068

EditingTools 2.1

Now you can have a slick full-screen multi-file text editor for only $35. Enjoy the ease of editing many files at once and the freedom of moving text among files. File sizes are limited only by memory. With an easily reconfigurable keyboard and a convenient multidirectory DOS shell. Add $4 for s/h.

Dr. Jiann Jou
P.O. Box 460969
Garland, TX 75046
(214) 495-8862

SCREEN MANAGER

SAVE TIME! Powerful Screen Designer and Data Entry Manager Increases Your Productivity! Interfaces to most languages. BASIC, FORTRAN, COBOL, C, PASCAL, PLM86, ASM. Not a Code Generator! No Royalties.

$79
VISA/MC

The West Chester Group
P.O. Box 1304
West Chester, Pa 19380
(215) 644-4206

FREE DEMO DISK

CIRCLE 285 ON READER SERVICE CARD

StruBAS

Structured BASIC Development System
QuickBASIC - IBM BASIC 2.0

Design screens in minutes with powerful screen designer using any editor. Modify most features without recompiling. Interface to Btrieve, ISAM, and Screens with single commands. Offers program generators, menu utility, subroutine library, utility programs, network support thru Btrieve, and preprocessor to extend BASIC and enhance structure.

NOT COPY PROTECTED

Site and Corporate licenses available.
LANEY SYSTEMS, INC.
3 Office Park Drive, Ste. 100
Little Rock, Arkansas 72211
501-225-7755
VISA/MC ACCEPTED $495/$5 demo.

FREE 30 DAY TRIAL

CIRCLE 286 ON READER SERVICE CARD

M Street Software

SCRUTINY
Advanced Symbolic Debugger
for all MS-DOS computers

M Street Software
214-827-4908

5400 E. Mockingbird, Suite 114
Dallas, Texas 75206

Information also available via our 24-hour 300/1200 modem: 214-669-1882

CIRCLE 287 ON READER SERVICE CARD

Productivity Tools

Software Revision Management System

SRMS™ is a full featured version control system featuring:

- 10 Integrated Utilities with user shell
- Capability for hundreds of versions/library
- Merge utility resolves parallel effort conflicts
- Report Generation Utilities
- Typeset documentation and much more!

Version 3.0 $185

QMAKE™

- Program generation utility patterned after UNIX make to aid in rebuilding sytems
- Recompiles only routines necessary
- Support for macros and multiple entry points
- Integrates fully with SRMS

Version 1.2 $99

TXTTOOLS

- QDIFF - Windowed File Difference Utility
- QSE - Quilt Stream Editor
- QSRCH - Like UNIX GREP

$85

Quilt Computing
7048 Stratford Road
Woodbury, MN 55125
(612) 739-4650

CIRCLE 283 ON READER SERVICE CARD

One-stop ROM Shop

When your application calls for ROM support, contact the professionals at ALDIA SYSTEMS. We've been supporting firmware developers for over two years with excellent software development tools and technical support. Don't settle for imitations, call the company who brought you ***PC-LOCATE,*** the original PC locater.

PC-LOCATE: Produce ROM-able code from your ".EXE" files. PC-LOCATE assigns physical addresses to the re-locatable image based on user inputs. PC-LOCATE support the entire Intel processor family including : 8086, 8088, 80186, 80188 and 80286.

PC-PROMPAK: A PROM/ROM expansion board for IBM and IBM-compatible computers. PC-PROMPAK provides up to 384Kbytes of non-volatile expansion memory and supports most 28-pin JEDEC devices including EPROMS, EEPROMS and Static RAMs.

PC-ROMDRIVE: Create a "Diskless PC" that can include MS-DOS and your application program. "Autoexec.bat" files are supported for automatic program execution.

ALDIA SYSTEMS, Inc.
P.O.Box 37634 Phoenix, Az. 85069
(602) 866-1786

CIRCLE 284 ON READER SERVICE CARD

Turbo Programmers

15 MINUTES = 200 HOURS with new *turboMAGIC* code generator.

Input forms and help windows up to 66 lines long. Scrolling within framed windows. Pop-up menus. Pull-down menu systems. And **much** more!

"turboMAGIC is the Cadillac of prototyping tools for Turbo Pascal."
Jim Powell,
COMPUTER LANGUAGE, June 87

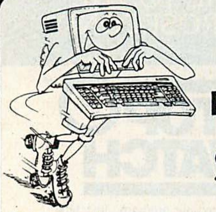
"Fast automatic updating of dependent fields adds flair to your input screens. ...turboMAGIC will be a blessing for programmers who would rather not write the user interface for every program."
Neil Rubenking,
PC MAGAZINE, 24 Feb. 87

Order your MAGIC today! Just $99. Call 800-225-3165. Money Back Guarantee. Requires 256K IBM PC compatible.

turbo magic™

Sophisticated Software Inc.
6586 Old Shell Road
Mobile, AL 36608
205-342-7026

CIRCLE 288 ON READER SERVICE CARD



NEW! TEXT EDITOR

$89

Before you buy an expensive text editor, look what you can do with ***FAST-EDI*** :

- Edit up to ten large files at once
- Use up to six windows
- Cut and paste between files
- Move, Copy, Delete, Save and Print blocks
- Enter DOS without exiting; edit output
- Perform full Search and Replace
- Define Macros . . . Plus lots more!

With ***FAST-EDI*** it's all fast and easy.

Comes with tutorial and moneyback guarantee.

System Vision Corp.
P.O. Box 281166
San Francisco, CA 94128
415/355-7308

CIRCLE 289 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Programmers Tools

PROFESSIONAL PROLOG TOOLS

A power packed collection of invaluable predicates for speeding up AI application development. Uncompiled code supplied with copious examples. No royalties. (Requires Turbo Prolog).

PRICE $79 VISA/MC/CHECK
30 day money back guarantee.
DRYDEN SOFTWARE
P.O. Box 238
Santa Barbara, CA 93102
(805) 683-1650

What is Turbo GhostWriter?

An Application Generator for Turbo Pascal™ Complete Database in Minutes! Draw & paint screens with our tools. Tell Turbo GhostWriter to retrieve your data. * Screen Editor/Painter * FABS PLUS B - Tree File Manager * Multiple fields/Key * Multiple Keys/ index. Application Generator produces Context sensitive Help & Documentation. All for only $200! 60-day money back guarantee. MC/VISA/PO/COD - Company Check.
Info/order: 800-227-7681; In NC (919) 782-7703
ASCII - Automated Software Concepts Int'l. Inc.
3239 Mill Run
Raleigh, NC 27612-4135

MULTI-TASKING TOOLKIT

Write your 'C' programs with cooperating, concurrent tasks using INTERWORK. Very useful for parallel programming, simulation, and real-time applications. INTERWORK supports more than 100 tasks, provides inter-task communication facilities, and handles DOS interrupts. PC/DOS version $129, Xenix $159, Unix $249.
Block Island Technologies
13563 NW Cornell Road, Suite 230
Portland, Oregon 97229-5892
(503) 241-8971

MEASURE EXECUTION TIME TO THE MICROSECOND

STOPWATCH™

- No mods to your program. Just type |CTL|-M instead of |ENTER| to measure execution time to the next display output.
- Use with MS debuggers to measure time for your program from "go" to breakpoint.
- Independent of CPU type and clockspeed.
- Disk I/O time broken out separately.

Only $59.95 + 3.50 s&h + 6% NJ tax.
MC/VISA.
CUSTOM REAL-TIME SOFTWARE, INC.
P.O. Box 1106, West Caldwell, NJ. 07007
U.S. 800-345-0167 • NJ 201-228-7623

™STOPWATCH is a Trademark of Custom Real-Time Software Inc.

CIRCLE 379 ON READER SERVICE CARD

Public Domain

IBM PUBLIC DOMAIN

Send two stamps for free catalog. We have games/word processors/data bases/spread sheets/graphics/inventory control budget programs/checking/artificial intelligence and much more. For home or business. All programs DDDS high quality disks. Only $3.00 per disk. Hundreds to choose from.
Echo Disk Copy
P.O. Box 50132
Mobile, AL 36605

TURBO PASCAL $2.50 - disk

TSS, the BBS-by-mail. 120+ disks of compressed files. Membership fee ($25) incl starter pkg + 2 FREE disks with 1st order, Cat list $5. Anniversary sale: $250 for full membership PLUS full, library thru disk 150. Visa/MC/COD + $5. Non U.S. + $10 U.S. funds only. Call our online BBS (617) 545-9131. All files online!
TURBO SOURCE SEARCH
P.O. Box 876
Scituate, MA 02066
(VOICE) 617-545-6677

Scientific

SCI/ENG GRAPHICS

OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

SCI-GRAF and SCI-CALC

SCI-GRAF produces huge hi-res graphs thru easy menus or linkable C libraries. Supports log scales, error bars, point labeling, screen and printer output. SCI-CALC is a pop-up calculator with complete expression editing, scientific, statistical, and logical functions. Requires IBM PC compatibility. Prices start at $79.95.
Microcomputer Systems Consultants
32 West Anapamu, Suite 190
Santa Barbara, CA 93101
(805) 963-3412

FAST FOURIER TRANSFORMS

* Subroutine for integer data: VERY FAST. 1024 points in 109 millisec on 8 MHz AT.
* Subroutine for floating-point: ACCURATE. 1024 points in 700 mS with 8 Mhz 80287.
* Call from MS FORTRAN, C, QuickBASIC, Turbo Pascal, Lattice C, assembler. GREAT DOCS.
* Check or PO: $175 ea/ $275 for both.
J.W. Hartwell & Associates
Route 4 Box 1540
Hillsborough, NC 27278
(919) 732-7951

Scientific

NUMERICAL C SOFTWARE

Computationally stable numerical routines for C software developers. LINLIB contains complex arithmetic, solutions to equations, approximations, LU, QR, Cholesky factors of matrices, least squares solutions, eigenvalues. LINLIB has splines, B-spline routines, spline interpolation, spline approximation of data. $165.
INFORMATION AND GRAPHIC SYSTEMS
15 Normandy Court
Atlanta, GA 30324
Call (404) 231-9582

Security Devices

BREAKTHROUGH IN SECURITY!

Ultimate protection against software piracy! Unlike current schemes, which can be bypassed by debugging tools, COMBITRON I utilizes realtime dynamic interaction between the protected software and a hardware unit through address and data scrambling. Virtually unbreakable. Available for PC/XT/AT/Compatibles and PS/2.
T LOG Technologies Corp.
9 Bentley Rd.
Plainview, NY 11803
(516) 932-9057 TELEX 650-276-0098

EVERLOCK COPY PROTECTION

Designed for user-transparency, clone compatibility & strength. It features:

- no need for damaged media or I/O plugs
- supports all Hard & Floppy disk formats
- file-server networks supported
- variable number of installs (0-99)
- create demos with remote unlock option
- allows protected upgrades by modem/BBS

$195 Starter Kit or $495 with NO meter counts. Free info & demo disk available.

AZ
Az-Tech Software, Inc.
305 East Franklin
Suite A4A
Richmond, MO 64085
(800) 227-0644 . . . (816) 776-2700

CIRCLE 291 ON READER SERVICE CARD

BIT-LOCK® SECURITY

Piracy SURVIVAL 5 YEARS proves effectiveness of powerful multilayered security. Rapid decryption algorithms. Reliable/small port transparent security device. PARALLEL or SERIAL port. Countdown and timeout options also available. KEY-LOK™ security at about ½ BIT-LOCK cost.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 922-6410 or 798-7683

Statistics

RATS! New Version 2.10

Full function regression, time series and forecasting program. Multiple regressions, including stepwise. Non-linear least squares. ARIMA models, transfer functions & intervention analysis. Exponential smoothing, spectral analysis, Kalman filter, model simulations and much more! High-quality graphics to screen, plotter, printer. $200-$300. Visa/MC.
VAR Econometrics, Inc.
P.O. Box 1818
Evanston, IL 60204-1818
(312) 864-8772; (800) 822-8038

PowerStat

An extremely powerful program with a good user interface — *PC Tech Journal*

Excellent analysis of variance routines — *Info World*

- Basic Statistics
- Regression
- Multivariate
- Counted Data
- Analysis of Variance

Version 1.04 US $395

Analytical Engineering Corporation
P.O. Box 9, Station P,
Toronto, Ontario, Canada M5S 2S6
(416) 960-3030

CIRCLE 292 ON READER SERVICE CARD

Terminal Emulation

PCBTAM Communications Access Method

General purpose binary synchronous access method for custom file transfer or terminal emulation on PC/XT/AT with IBM BSCA card.

- high performance, full featured
- object ($300.00) or source ($2,500.00) license
- link with ASM, Lattice or Microsoft programs
- Z-SIO version available

Symbiotic, Inc.
1035 Route 46 East
Clifton, NJ 07013
201-777-6454

CIRCLE 293 ON READER SERVICE CARD

Term/Comm Productivity Tool

PereLine emulates VT100, VT52, DG210/211, IBM3101 Vidtex and two connections at same time. Learn mode lets your PC create autologons and other scripts. Remote operations, exit to DOS while connected, customize your own menus and 5 file transfer protocols including batch file transfer Only $69.95
PEREGRINE DATA SYSTEMS INC.
5365 Baron Drive
San Jose, CA 95124
408-356-6105

TECH MARKETPLACE . . . The Comprehensive Guide to Products and Services for the MS DOS Market
For more information call **(212) 503-5115**

TERMINAL EMULATION

category continued on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Terminal Emulation

BARR/HASP INTELLIGENT RJE WORKSTATION

Hardware and software communications package for IBM PC, XT and AT. Simultaneously transmits data to host and receives output directly to MVS/JES2, MVS/JES3, VS/RSCS, and CDC/NOS, bypassing TSO and CMS. Emulates IBM 3777-2 and HASP on IBM 360/20. Line speed: 1,200 to 19,200 baud (56,000 bps on AT). Supports multiple high-speed printers beyond 2,400 lpm. (6,000 lpm on AT). Features: concurrent DOS, LAN support, printer forms control, plotter support, unattended operation, easy installation. $1,290 includes Hardware & Software.

BARR

BARR SYSTEMS, INC.
2830 NW 41st Street, Building M
Gainesville, FL 32606
(800)-BARR-SYS/(904) 371-3050

Utilities

DISK ACCELERATOR V2.0

DiskCache speeds up your hard disk access. Disk caching and ram disk in one package. Ram disk shares cache space. Transparent, flexible, configurable, no h/w changes. RAM, EMS, and AT extended memory versions incl. Not copy protected. VISA, MC, volume discounts. No PO's w/o prior approval. $49.00
DATAMORPHICS LTD.,
P.O. Box 820
Stittsville, Ontario, Canada KOA 3G0
Or call (613) 836-2670

HARD DISK EXPANSION!

Disk Manager allows the installation of any ST506 hard disk on PC,XT,AT and compatibles. Volumes up to 256mb! Menu driven/auto install, compatible w/ all vers of MS/PC DOS (does not modify DOS), up to 16 volumes, easy to use! $125+ ship. Ask about Novell product! Dealer inquiries invited.

ONTRACK COMPUTER SYSTEMS INC.

ONTRACK COMPUTER SYSTEMS, INC.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107 1-800-752-1333

Tech Marketplace...
Second Color Option Available
For More Information or to Place Your Ad
Call (212) 503-5115

Utilities

VIEW ANY DOS FILE

View, print, modify, or search the contents of any file. File Dump 2.0 lets you view a file continuously or as fixed length records. Record lengths up to 50K are supported. Select any record. Search and update in hex and ASCII. Prints mainframe-style dumps. Search & modify. EXE files easily. $49.95+$4 s/h. CT add 7.5
LASOR Computer Specialists
P.O. Box 5137
Rocky Hill, CT 06067
(203) 721-0261

SPEED UP YOUR HARD DISK!

Vopt™ DISK ORGANIZER

For fast, efficient, time-saving use, organize your hard disk with *Vopt*!

Vopt consolidates the fragmented files that slow down and wear down your disk. Now your hard disk can find what it's looking for — all in one place!

Vopt is fast! Initial organization averages under 5 minutes! Daily runs from the AUTOEXEC file average under 1 minute!

Vopt includes — *Vmap* for graphic display of HD utilization, *Vseek* for HD seek times, *Vmarkbad* to flag bad clusters, plus much more!

Get *Vopt* and get going!

ONLY $49.95
Add $3 shipping & handling
CA residents add 6% sales tax

GOLDEN BOW SYSTEMS

2870 Fifth Avenue
Suite 201
San Diego, CA 92103
619/298-9349

Vopt is a trademark of Golden Bow Systems

CIRCLE 295 ON READER SERVICE CARD

Introducing

Quaid Analyzer

the tool that created CopyWrite

Now you can debug your own programs with a professional quality debugger - the one that unraveled every form of copy-protection used on the PC.

With the Quaid Analyzer, you can:

- ☐ See occurrences of any interrupt, with its meaning shown on the screen.
- ☐ View memory as text or instructions, scrolling as easily as you do with an editor.
- ☐ Run until a memory location or I/O port is changed.
- ☐ Protect your hard disk from accidental destruction.
- ☐ Analyze software without the source, even when it uses countermeasures to thwart tracing.
- ☐ See all stages of the boot load.

We kept the Quaid Analyzer off the market to avoid helping publishers with copy-protection. Now that copy-protection is gone, we can sell it to you.

The Quaid Analyzer is a software tool occupying 100K bytes. It runs on any IBM PC and most MS-DOS systems without hardware modification.

Quaid Software Limited

$99 U.S.
All orders shipped at our expense within a day. All major credit cards accepted.

To order

Call **(416) 961-8243**
or write to:
45 Charles St. East
Third Floor, Dept. 602
Toronto, Ontario. M4Y 1S2

Ask about Disk Explorer the program that takes over where Quaid Analyzer leaves off.

CIRCLE 294 ON READER SERVICE CARD

Compress Your Data 10X FASTER!!

- PKARC & PKXARC can compress your files even smaller and up to TEN TIMES FASTER than the other ARChive program.
- **"PKARC/PKXARC is the system to use."**
  -Dr. Dobbs Journal of Software Tools

Only $45 + 3.50 s/h. WI res. add 5% tax.
7032 N. Ardara Ave., Glendale, WI 53209 (414) 352-3670

PKware Inc.

CIRCLE 296 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

ASCII FILE COMPARISON

Side by side comparison in sliding windows; printed record of program updates. Highlights labels, subroutine names, line numbers. Custom or standard label, statement and remark delimiters. Matches out-of-sequence sections. $35.00 includes shipping.

Opus-1 Systems, Inc.
300 108th Ave. SE
Bellevue, WA 98004
(206) 454-4879

THE COMPLETE DISK UTILITY

If it's on a disk or diskette, Disk Explorer will find and display it. Disk Explorer, the complete disk utility: Recovers files. Displays, changes, creates and compares sectors. Creates tracks that can't be duplicated by DOS, locates the precise end of a track, creates specialized track formats. Allows viewing and moving of clusters. Reports on disk drive speed, increases the speed of reading data and much more. IBM PC, XT or AT, 128kb and one diskette drive. $75.00 U.S. Check/Credit card welcome.

Quaid Software Limited

QUAID SOFTWARE LIMITED
45 Charles St. East, Third floor
Toronto, Ontario M4Y 1S2
(416) 961-8243

RECORD/PLAYBACK KEYSTROKES

TOTAL RECALL records an unlimited number of keystrokes and plays them back after a failure or mistake. Also for demos, running interactive programs in batch, or repeating keystrokes. 3 playback speeds plus ACTUAL for timing sensitive programs. Built-in edit. PC/XT/AT/Compat. $79 VISA/MC 30 day guarantee.

Computer Foundations
2604 Elmwood Avenue, Suite 320
Rochester, NY 14618
(716) 586-9756

LIMSIM

Expanded Memory Simulator for the PC/AT and compatible 286 machines. Use the extended memory you already have as Lotus style Expanded Memory. Fully supports EMS version 3.2. Requires 70k of conventional memory. $50 ($75 with assembler source) plus $5 s/h. 30 day money back guarantee. Visa/MC accepted.

Larson Computing
1556 Halford Ave. #142
Santa Clara, CA 95051
(408) 737-0627

XT/AT HARD DISK DIAGNOSTICS!

Disk Manager Diagnostics performs extensive tests on your ST412/506 hard disks. Areas tested are: Controller, data write/read, seek test, automatic error correction(ECC), random reads and media defects. Interactive help. Excellent error detection and isolation. $49.95 + ship. VISA/MC accepted.

ONTRACK COMPUTER SYSTEMS INC.

ONTRACK COMPUTER SYSTEMS, INC.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107 1-800-752-1333

AT's DON'T NEED 360KB DRIVES

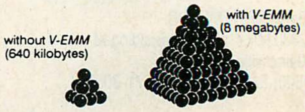
The 1.2MB drive has long been known to **READ** but **NOT** reliably **WRITE** on 360KB floppies. With **"CPYAT2PC"** 1.2MB drives **CAN** reliably **WRITE** 360KB floppies saving a slot for a second hard disk or backup tape. **"CPYAT2PC"** (Not Copy Protected) offers the preferable **SOFTWARE SOLUTION.**

- **NO** software or hardware modification
- A 360K drive is **NOT** required
- **"CPYAT2PC"** program **MAY** reside on hard disk
- Runs on IBM PC/AT and COMPATIBLES i.e. Compaq Deskpro 286/386, AT&T 6300+, HP Vectra, Sperry PC/IT, Tandy 3000

Only $79.00 + $4.00 S/H VISA, MC, COD, UPS-B/R
ORDER TOLL FREE 1-800-621-0851 XT777
TELEX EZLINK 62873089
Dealer Inquiries Invited
MICROBRIDGE COMPUTERS
655 Skyway #125
San Carlos, CA
CA 415-593-8777
NY 212-334-1858

CIRCLE 298 ON READER SERVICE CARD

Ammunition to conquer your biggest problems!



- The **Virtual Expanded Memory Manager** provides your EMS applications with up to eight megabytes of expanded memory.
- It turns PC/AT extended memory into expanded memory.
- It turns hard disk storage into expanded memory.
- It's compatible with most EMS applications, including 1-2-3, Symphony, and Reflex.
- *Only $89.95. 30-day money-back guarantee.*

V-EMM™

Fort's Software
P. O. Box 396
Manhattan, KS 66502
(913) 537-2897

CIRCLE 300 ON READER SERVICE CARD

FREE SPACE ON FULL DISKS WITH squish®

The ***Automatic*** File Compressor

Get megabytes of extra disk space ***without*** the expense or effort of installing a new hard disk.

Squish even lets other software (dBASE III, R:base, etc.) read or update compressed files ***while they stay compacted on disk... without any software changes! That's right, without expanding one byte on disk!***

A must for databases... compress them up to **90%!** Reduce text files, spreadsheets, etc. up to 60%.

"Ideal for... large data files"
PC Magazine 2/24/87

"You owe it to yourself... check this one out" ***NYPC Newsletter Feb 87***

PC/XT/AT•DOS 2.0 & up•40K resident
Order today! Just $79 + $5 S/H:

SunDog Software Corporation
(718) 855-9141
264 Court St, Brooklyn, NY 11231

CIRCLE 297 ON READER SERVICE CARD

Compress your data into 1/10th the space!

Introducing ARC. It's used to create and maintain data file archives for computers operating under any DOS system.

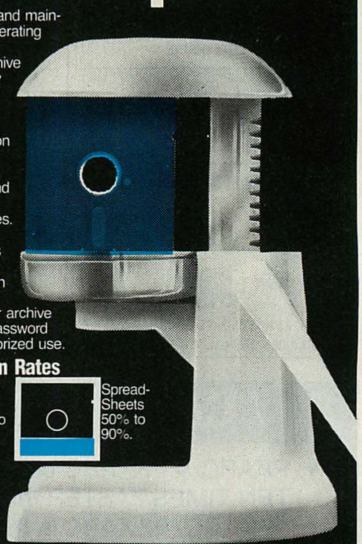
But it does something that other archive and library utilities can't. It automatically squeezes the files being saved so they take up less space. Like a can of concentrated orange juice.

From 20% to 90% less, depending on the kind of data being saved!

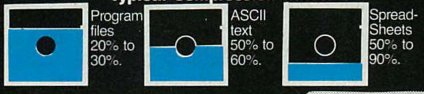
So there's more room to store data, no matter what media it's stored on! And that's like giving a shot of vitamin C to your savings on equipment and supplies.

This compressed data can be transmitted over telephone lines in a lot less time than it takes to transmit uncompressed data. So you can beat the high cost of phone bills to a pulp, as well.

ARC has a full range of functions for archive creation and maintenance. Including password encryption to protect data from unauthorized use.



Typical Compression Rates



sea

System Enhancement Associates • 21 New Street, Wayne, NJ 07470 • (201) 473-5153

CIRCLE 299 ON READER SERVICE CARD

PC TECH JOURNAL MARKETPLACE is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches.

Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $170 per issue ($510 total).

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

TECH MARKETPLACE . . .
The Home of the Power Buyer

SOFTWARE/DATA BASE MANAGEMENT

MISCELLANEOUS/ BAR CODING

BUSINESS OPPORTUNITIES

PUBLICATIONS

INDEX TO ADVERTISERS

INDEX TO PRODUCTS

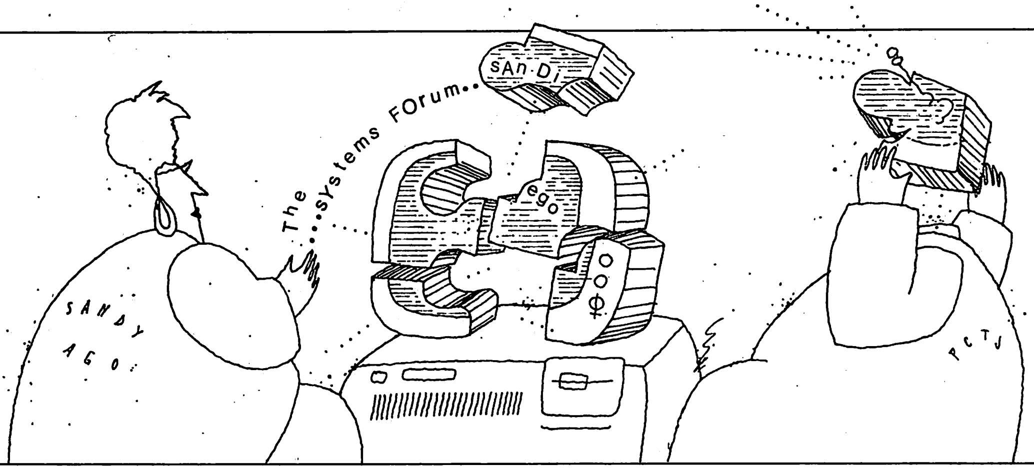


ILLUSTRATION • DAVID POVILAITIS

SEPTEMBER

September 1–3
PC EXPO
New York, NY (PC EXPO) *Contact:* Jim Mion, PC EXPO, 333 Sylvan Ave., Englewood Cliffs, NJ 07632; 800/922-0324; in New Jersey, 201/569-8542

September 14–18
Electronic Printer and Publishing Conference
Miami, FL (CAP International) *Contact:* CAP International, One Snow Road, Marshfield, MA 02050; 617/837 -1341

September 21–23
Conference on Software Maintenance
Austin, TX (NBS, DPMA, and IEEE-CS) *Contact:* Roger Martin, National Bureau of Standards, Bldg. 225, Rm. B266, Gaithersburg, MD 20899; 301/921-3545

September 23–25
***PC Tech Journal* Systems Forum**
San Diego, CA (*PC Tech Journal*) *Contact:* Marti Cunha, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia MD 20144; 301/740-8300

September 23–25
Writing Better Computer Software Documentation for Users
Atlanta, GA (Georgia Institute of Technology) *Contact:* Deidre Mercer, Dept. of Continuing Education, GIT, Atlanta, GA 30332; 404/894-2547

September 28–October 1
Conference on Electronic/ Desktop Publishing
San Francisco, CA (National Computer Graphics Association) *Contact:* NCGA, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031; 800/225-6242; 703/698-9600

September 29–October 2
INFO 87
New York, NY (Cahners Expositions Group) *Contact:* Frank Fazio, Show Manager, INFO 87, 999 Summer St., Stamford, CT 06905; 203/964-8287

OCTOBER

October 4–8
OOPSLA '87
Kissimmee, FL (ACM SIGPLAN) *Contact:* Object Oriented Programming: Systems, Languages, and Applications Conference; ACM, 11 W. 42nd St., New York, NY 10036; 212/869-7440

October 5–7
National Connectivity Symposium
Chicago, IL (Digital Consulting Associates, Inc.) *Contact:* Seminar Services Department, 6 Windsor Street, Andover, MA 01810; 617/470-3870

October 5–8
ASPLOS-II
Palo Alto, CA (ACM SIGPLAN) *Contact:* Architectural Support for Programming Languages and Operating Systems Conference, ACM, 11 W. 42nd St., New York, NY 10036; 212/869-7440

October 6–7
Data Security and Control: A Management Overview
New York, NY (New York University) *Contact:* NYU, School of Continuing Education, Seminar Center, 575 Madison Ave., New York, NY 10022; 212/580-5200

October 13–15
PC EXPO/Chicago
Chicago, IL (PC EXPO) *Contact:* Jim Mion, PC EXPO, 333 Sylvan Ave., Englewood Cliffs, NJ 07632; 800/922-0324; in New Jersey, 201/569-8542

October 15–17
Northeast Computer Faire
Boston, MA (Interface Group) *Contact:* The Interface Group, Inc., 300 First Ave., Needham, MA 02194; 617/449-6600

October 19–23
Expert Systems in Government
Washington, DC (Mitre Corporation and IEEE-CS) *Contact:* IEEE-CS, 1730 Massachusetts Ave. NW, Washington, DC 20036; 202/371-0101

October 20–22
NetWorld '87
Dallas, TX (PC EXPO) *Contact:* Annie Zdinak, PC EXPO, 333 Sylvan Ave., Englewood Cliffs, NJ 07632; 800/526-3247; in New Jersey, 201/569-6409

October 25–27
Print Quality Seminar
Bedford, MA (Datek) *Contact:* Frank Stefansson, Datek Information Services, Inc., P.O. Box 68, Newtonville, MA 02160; 617/893-9130

October 26–28
CEPS/Fall
Boston, MA (Cahners Exposition Group and InterConsult, Inc.) *Contact:* Corporate Electronic Publishing Systems Conference, InterConsult, Inc., 48 Brattle Street, Cambridge, MA 20138; 800/255-7798; 617/547-0332

October 27–29
UNIX EXPO
New York, NY (National Expositions Company) *Contact:* NEC, 49 W. 38th St., Suite 12A, New York, NY 10018; 212/391-9111

October 28–30
AI/East '87
Atlantic City, NJ (Tower Conference Management) *Contact:* TCM, 331 W. Wesley St., Wheaton, IL 60187; 312/668-8100

NOVEMBER

November 2–4
DPMA Computer Conference
San Francisco, CA (Data Processing Management Association) *Contact:* DPMA, 505 Busse Hwy., Park Ridge, Il 60068-3191; 312/825-8124

November 3–5
COMDEX/Fall
Las Vegas, NV (Interface Group) *Contact:* The Interface Group, Inc., 300 First Ave., Needham, MA 02194; 617/449-6600

November 9–11
Symposium on Operating System Principles
Austin, TX (ACM SIGOPS) *Contact:* Les Belady, Microelectronics Computer Consortium (MCC), 9430 Research Blvd., Echelon Building 1, Suite 200, Austin, TX 78759; 512/834-3330

November 9–12
ICCAD-87
Santa Clara, CA (IEEE-CS) *Contact:* International Conference on Computer-aided Design, IEEE-CS, 1730 Massachusetts Ave. NW, Washington, DC 20036; 202/371-0101

November 11–13
Localnet '87
Los Angeles, CA (Online International) *Contact:* Online International, Inc., 989 Avenue of the Americas, New York, NY 10018; 212/279-8890

November 11–13
Optical Publishing and Storage
New York, NY (Learned Information, Inc.) *Contact:* Learned Information, Inc., 143 Old Marlton Pike, Medford, NJ 08055; 609/654-6266

DECEMBER

December 1–3
Optical Information Systems '87
New York, NY (Conference Management Corporation) *Contact:* Meckler Publishing, 11 Ferry Lane W, Westport, CT 06880; 203/226-6967

December 6–9
International Conference on Information Systems
Pittsburgh, PA (Society for Information Management) *Contact:* William D. King, Graduate School of Business, University of Pittsburgh, Pittsburgh, PA 15260; 412/648-1587

December 16–18
Microcomputer Graphics '87
New York, NY (Expoconsul International) *Contact:* Expoconsul International, Inc., 3 Independence Way, Princeton, NJ 08540; 609/987-9400

Last year, Advanced Logic Research introduced the first 80386 systems. Now we're introducing the ALR 386/2,™ the second generation.

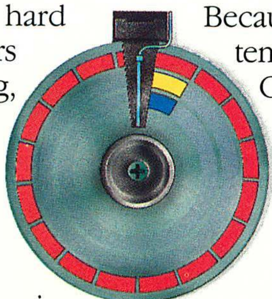
Until now, if you wanted three times the speed of an AT™ you paid about three times the price of an AT. Now Advanced Logic Research announces second-generation 386 systems—designed to do to the price barrier what our first generation did to the performance barrier. Annihilate it.

Starting at $1990, ALR's totally new 386/2 systems couple the power of 32-bit processing with true 32-bit memory. Even the system and graphics BIOS are implemented in a 32-bit architecture. That simply means your applications will run faster on a 386/2 than any other available computer. And ALR 386/2 systems let you use all the peripherals, graphics, enhancements and applications developed for the most widely adopted computer operating environment in history.

Which makes you wonder why others want thousands more for less flexible, first-generation 386 systems.

How to run circles around the competition.

Sure, Compaq and IBM use the fastest available hard disks and controllers with 1:1 interleaving,



just like Advanced Logic Research. But they don't buffer a full 17-sector hard disk track, settling for sector by sector buffering. Our way makes the fastest even faster where it counts—in the real world.

And with up to 2 MB of RAM on the motherboard, you get flexibility with your power.

Picture this.

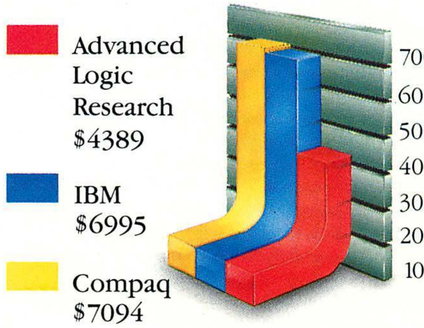
Naturally the raw speed of 80386 means the 386/2 series make great EGA graphic workstations for CAD/CAM. Or choose enhanced EGA™ or GA 786™ graphics from ALR and a variety of sources and get the most advanced resolutions available.

The 386/2 series makes the best use of floor or desk space.

You can even run applications without memory limitations. Because all enhanced ALR systems include the Phoenix Control/386 Software utilities with 32 bit disk caching, Vdisk and EMS/EEMS Software.

Speed to burn. Without having money to burn.

Read the reviews and compare the 386/2 to the others. Then compare more. If you find more power, flexibility and quality somewhere else, buy somewhere else. You won't find a more competitive price anywhere else.



The ALR 386/2 Model 40 with EGA adapter is similar to the IBM Model 80-041 and Compaq Deskpro 386 Model 40 with EGA adapter. Except for a lower price and twice their standard RAM.

Advanced Logic Research. Faster, first.

Advanced Logic Research got its start designing high-performance microcomputers for customers that demanded more power than they could get off the shelf. We designed one of the first IBM PC-compatibles. Developed the enhanced performance AT-compatible *PC Magazine* called "...the most judicious choice..." And introduced the first 386 system, which *PC Tech Journal* said "...brings up-to-date technology to affordable 386 systems."

Our complete line of performance XT and AT-compatible systems offers the advanced technology of one of the world's most experienced PC

FREE
20-MHz OPTION
ON 386/2 Model 130
systems until November 1

design teams. And the cost-effective quality of Singapore-based manufacturing. So you simply get more computing power for less money. From a growing international network of ALR dealers chosen for their ability to deliver full-service and support.

Call Advanced Logic Research today for the name of your local ALR dealer. And discover the price of power isn't as high as it used to be.

Performance Specifications

386/2 Model 10 — $1990

- ALR-designed system board
- 1 MB 32-bit RAM, expandable to 2 MB on system board
- 1.2 MB floppy disk drive
- Serial and parallel ports
- 8 expansion slots
- 101-key keyboard
- 16-MHz 80386 processor (20-MHz optional)
- 80387 support

386/2 Model 40 — $3990

- ALR-designed system board
- 2 MB 32-bit RAM
- 40 MB, 30 ms, or less, access time hard disk drive/ 500 Kbs/s trans rate
- EMS and EEMS software
- 1.2 MB floppy disk drive
- Serial and parallel ports
- Desktop or floormount
- 8 expansion slots
- 101-key keyboard
- 16-MHz 80386 processor (20-MHz optional)
- 80387 support
- 32-bit Vdisk and disk caching software

386/2 Model 80 — $4690

- ALR-designed system board
- 2 MB 32-bit RAM
- 70 MB, 30 ms, or faster, access time hard disk drive/ 500 Kbs/s trans rate
- EMS and EEMS software
- 1.2 MB floppy disk drive
- Serial and parallel ports
- Desktop or floormount
- 8 expansion slots
- 101-key keyboard
- 16-MHz 80386 processor (20-MHz optional)
- 80387 support
- 32-bit Vdisk and disk caching software

386/2 Model 130 — $7299

- ALR-designed system board
- 2 MB 32-bit RAM
- 130 MB, 30 ms, or faster, access time hard disk drive/ 500 Kbs/s trans rate
- EMS and EEMS software
- 1.2 MB floppy disk drive
- Serial and parallel ports
- 8 expansion slots
- 101-key keyboard
- 16-MHz 80386 processor (20-MHz optional)
- 80387 support
- 32-bit Vdisk and disk caching software

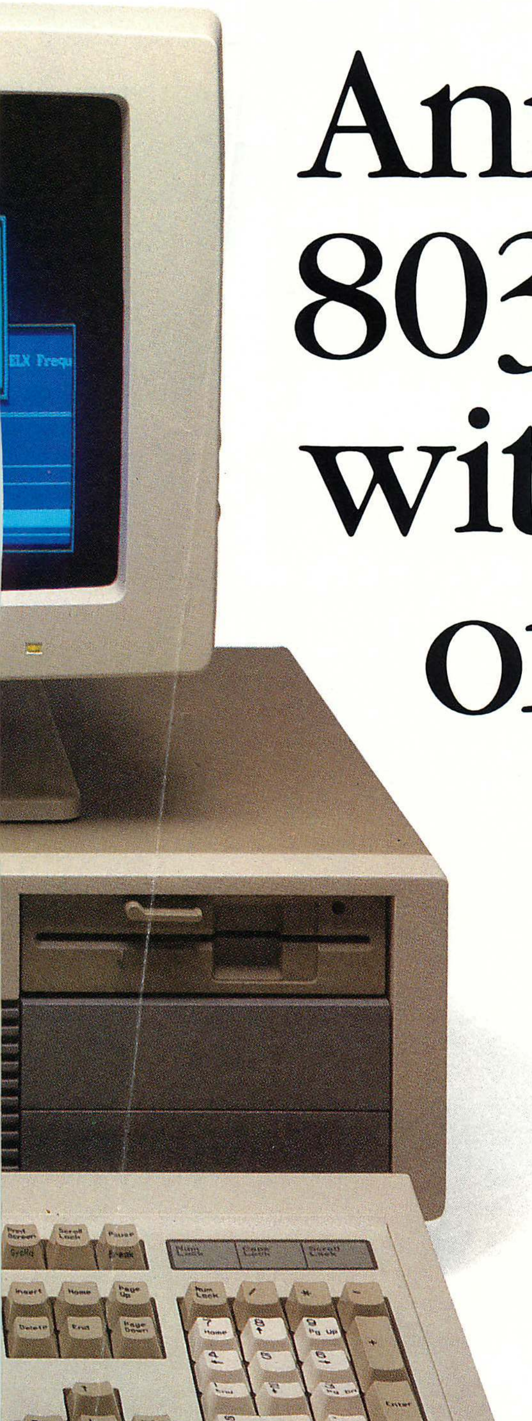
Enhancements

A complete range of enhancements, including 4 MB 32-bit RAM and multifunction products as well as additional I/O options are available. 640 x 480 pixel AV EGA with 16 colors and GA 786 graphics adapters available third quarter.

Monitor not included

Advanced Logic Research, Inc.
10 Chrysler, Irvine, CA 92718 (714) 581-6770
FAX: (714) 581-9240 TELEX: 5106014525, Answer back Advanced Logic
From Asia or Europe call ALR/Wearnes Technology
Phone: (65) 2592521 TELEX: RS38113WRNTEC

Circle Reader Service No. 116 for Re-seller, No. 121 for End-user.



Announcing 80386 power with 1 MB of 32-bit RAM for less than $2,000.

And a blazing new 20-MHz system for less than $2,500.